

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería de Tecnologías y Servicios de  
Telecomunicación (EUR-ACE®)**

## **TRABAJO FIN DE GRADO**

**Diseño de un regulador digital universal implementado en  
FPGA para convertidores de potencia**

**Jorge Benedicto Ridaura  
Tutor: Alberto Sánchez González  
Ponente: Ángel De Castro Martín**

**Julio 2018**



# **Diseño de un regulador digital universal implementado en FPGA para convertidores de potencia**

**AUTOR: Jorge Benedicto Ridaura**

**TUTOR: Alberto Sánchez González**



**Dpto. de Tecnología Electrónica y de las Comunicaciones**

**Escuela Politécnica Superior**

**Universidad Autónoma de Madrid**

**Julio de 2018**



# Resumen

Un regulador es un sistema que se encarga de realizar un control sobre otro sistema, o planta, con el fin de conseguir un valor de referencia pedido por el usuario. Antiguamente, este control se realizaba analógicamente, pero gracias a las nuevas tecnologías ahora puede ejecutarse digitalmente, lo que provoca numerosas ventajas como reguladores más robustos, con capacidades de monitorización, reconfiguración, etc.

Una de las características importantes de los reguladores es el tipo de aritmética que utilizan, obteniendo dos grandes grupos: coma fija y coma flotante. Mientras que la coma flotante es muy versátil pero costosa en términos de velocidad y recursos, la coma fija permite sistemas optimizados, pero poco flexibles. En particular, la coma fija no permite que la posición de la coma pueda ser movida para poder adaptarse a números alejados entre sí. Por ello, el propósito de este trabajo consistirá en la creación de un regulador digital universal implementado en coma fija, pero con capacidad de ser configurado para adaptarse a cualquier regulador pedido. Para la elaboración de dicho regulador digital se utilizará una FPGA y la principal ventaja de la propuesta es que no será necesario resintetizar el código para poder realizar dicha configuración.

Un inconveniente asociado a la utilización de dicha placa es la necesidad de recurrir a una programación en VHDL más complicada. Estos problemas de implementación serán solventados mediante el uso de la coma fija, con capacidad de parametrización, lo que implica que el regulador deberá contar con la capacidad de adaptarse a los nuevos valores sin resintetizar el código de la FPGA. Por tanto, el regulador será configurado únicamente mediante un ordenador y una sencilla aplicación.

Con este Trabajo Fin de Grado, que está orientado a mejorar la docencia de Sistemas de Control, el estudiante podrá probar multitud de valores para el regulador y conseguir el que mejor se adapte a las características del circuito presente, realizando continuas pruebas a tiempo real. Esto permitirá que el usuario entienda mejor los sistemas de control y sus dispositivos, sin precisar de conocimientos elevados de programación en VHDL.

Por ende, la configuración del regulador universal será sencilla para poder probar de forma fácil y rápida diferentes reguladores diseñador por los estudiantes. A lo largo del presente trabajo se diseñará e implementará una aplicación de ordenador (mediante lenguaje Java) que se comunicará con dicho regulador, permitiendo configurarlo de forma gráfica. El trabajo concluye con una serie de resultados experimentales de sendos reguladores para un circuito eléctrico de tipo resistencia-condensador y para un circuito resistencia-bobina-condensador, y las consecuentes comprobaciones del correcto funcionamiento del circuito creado, comparándolo con los resultados teóricos.

## Palabras clave

Control digital, coma fija, field-programmable gate array, coma flotante, coma fija parametrizable, regulador digital.



# Abstract

A regulator is a system which is responsible for performing a control on another system, or plant, with the aim of getting a reference value demanded by the user. Once upon a time, this control was done analogically, but thanks to the new technologies now there can be executed digitally, which provokes numerous advantages such as robust regulators, which capacities of monitoring, reconfiguration, etc.

One of the important characteristics of regulator is the type of arithmetic that they use, obtaining two big groups: fixed point and floating point. Whereas the floating point is very versatile but costly in terms of speed and resources, the fixed point allows optimized, but slightly flexible systems. In particular, the fixed point does not allow that the position of the comma could be moved to be able to adapt to numbers removed between yes. For it, the intention of this work will consist of the creation of a digital universal regulator implemented in a fixed point, but with aptitude to be formed to adapt to any asked regulator. For the production of the above mentioned digital regulator a FPGA will be use and the main advantage of the offer is that will not be necessary to re-synthesize the code to be able to realize the mentioned above configuration.

A disadvantage associated with the utilization of the mentioned above plate is the need to resort to a programming in more complicated VHDL. These problems of implementation will be solved by means of the use of the fixed point, with capacity of customization, which implies that the regulator will have to possess the aptitude to adapt to the new values without re-synthesizing the code of the FPGA. Therefore, the regulator will be formed only by means of a computer and a simple application.

With this Work End of Degree, which is orientated to improve the teaching of the subject Systems of Control, the student will be able to prove multitude of values for the regulator and getting better the one that adapts to the characteristics of the present circuit, performing you continue try at real time. This will allow that the user should understand better the systems of control and his devices, without being necessary high knowledge of programming in VHDL.

Thus, the configuration of the universal regulator will be simple to be able to try easy and quick form different regulatory designer for the students. Along the present work an application of computer will design and implement (by means of language Java) that will communicate with the mentioned above regulator, allowing to form it of graphical form. The work concludes with a series of experimental results of two regulators for an electrical circuit of resistance - capacitor type and for a circuit resistance - inductor-capacitor, and the consequent checks of the correct functioning of the created circuit, comparing it with the theoretical results.

## Keywords

Digital control, fixed point, field-programmable gate array, floating point, parametrizable fixed point, digital regulator.





## ***Agradecimientos***

*En primer lugar quería agradecer a mi tutor, Alberto Sánchez, por brindarme la oportunidad de realizar este trabajo con él. También agradecerle el apoyo, las correcciones y las constantes ayudas que me ha aportado durante este último año. También quería agradecer a Ángel de Castro por hacer que me interesase por los sistemas de control desde aquella primera clase de tercero.*

*Por supuesto agradecer a mi familia todo el apoyo recibido estos cuatro años. Gracias a mis padres por hacer posible terminar esta etapa después de tanto esfuerzo y por siempre estar en el lugar correcto cuando más lo necesitaba. Mención especial a mi hermana, por ayudarme a realizar cualquier tarea, ya sean memorias, correos o trabajos creativos, y por estar siempre ahí cuando más la necesitaba. Gracias a mi padre por hacerme del Atleti, puesto que he aprendido que siempre hay que luchar hasta el final y que con esfuerzo y sacrificio cualquier objetivo es posible.*

*También dar las gracias a mi novia Cecilia, la cual ha hecho que este último año y medio estar en la universidad haya sido como estar en el paraíso, por conseguir sacarme todos los días una sonrisa y por hacerme mejor persona cada día. El Máster no será lo mismo sin ti.*

*Por último, no puedo olvidarme de todos los compañeros y amigos que me han apoyado cada día. Por supuesto, una mención especial a Eric por llevar más de 7 años trabajando en equipo codo con codo; y a Nacho por esas charlas interminables de fútbol y las ayudas continuas.*



# ÍNDICE DE CONTENIDOS

<b>1 Introducción</b>	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Organización de la memoria	2
<b>2 Estado del arte</b>	3
<b>3 Diseño</b>	5
3.1 Reguladores PID	5
3.2 Lazo cerrado	6
3.2.1 Ecuaciones en diferencias	7
3.2.2 Diseño del regulador	9
3.3 Adaptación a la FPGA	10
3.4 Coma fija parametrizable	10
<b>4 Integración del controlador</b>	15
4.1 Controlador	15
4.1.1 UART	16
4.1.2 Gestor PWM	17
4.2 Interfaz gráfica	18
<b>5 Integración, pruebas y resultados</b>	21
5.1 Circuito RC	21
5.2 Circuito RLC	26
<b>6 Conclusiones y trabajo futuro</b>	31
6.1 Conclusiones	31
6.2 Trabajo futuro	32
<b>Glosario</b>	35
<b>Anexos</b>	I
A    Capturas originales	I

# ÍNDICE DE FIGURAS

FIGURA 3-1. SISTEMA DE CONTROL CON PID.....	5
FIGURA 3-2. ACCIÓN DE CONTROL PROPORCIONAL .....	6
FIGURA 3-3. ACCIÓN DE CONTROL PROPORCIONAL-INTEGRAL .....	6
FIGURA 3-4. ACCIÓN DE CONTROL PROPORCIONAL-DERIVATIVA.....	6
FIGURA 3-5. ACCIÓN DE CONTROL PROPORCIONAL-INTEGRAL-DERIVATIVA .....	6
FIGURA 3-6. SISTEMA DE CONTROL DE LAZO CERRADO .....	7
FIGURA 3-7. FÓRMULA DE UN REGULADOR UNIVERSAL .....	7
FIGURA 3-8. EJEMPLO DESPLAZAMIENTO SEÑAL .....	8
FIGURA 3-9. DISEÑO DE UN REGULADOR PID EN VHDL.....	9
FIGURA 3-10. DISEÑO DEL REGULADOR COMPLETO .....	12
FIGURA 4-1. DISEÑO DEL SISTEMA COMPLETO .....	15
FIGURA 4-2. DISEÑO DEL GESTOR PWM.....	17
FIGURA 4-3. FORMULARIO DE LAS VARIABLES .....	19
FIGURA 5-1. CIRCUITO RC .....	21
FIGURA 5-2. FUNCIÓN DE TRANSFERENCIA DEL CIRCUITO RC .....	21
FIGURA 5-3. FUNCIÓN GENERAL REGULADOR RC .....	22
FIGURA 5-4. RESPUESTA AL ESCALÓN DEL CIRCUITO RC .....	22
FIGURA 5-5. SIMULACIÓN MODEL SIM CIRCUITO RC.....	23
FIGURA 5-6. SIMULACIÓN REAL CIRCUITO RC.....	23
FIGURA 5-7. SIMULACIÓN MODEL SIM CIRCUITO RC.....	24
FIGURA 5-8. SIMULACIÓN REAL CIRCUITO RC.....	24
FIGURA 5-9. SIMULACIÓN MODEL SIM REGULADOR CIRCUITO RC.....	25
FIGURA 5-10. SIMULACIÓN SIN DECIMALES EXCEL REGULADOR CIRCUITO RC.....	25
FIGURA 5-11. SIMULACIÓN CON DECIMALES EXCEL REGULADOR CIRCUITO RC .....	26

FIGURA 5-12. CIRCUITO RLC .....	26
FIGURA 5-13. FUNCIÓN DE TRANSFERENCIA CIRCUITO RLC .....	26
FIGURA 5-14. FUNCIÓN GENERAL REGULADOR RLC .....	27
FIGURA 5-15. RESPUESTA AL ESCALÓN CIRCUITO RLC .....	27
FIGURA 5-16. SIMULACIÓN REAL CIRCUITO RLC .....	28
FIGURA 5-17. SIMULACIÓN MODELSIM CIRCUITO RLC .....	28
FIGURA 5-18. SIMULACIÓN SIN DECIMALES EXCEL REGULADOR CIRCUITO RLC .....	29
FIGURA 5-19. SIMULACIÓN MODELSIM REGULADOR CIRCUITO RLC .....	29
FIGURA 5-20. SIMULACIÓN CON DECIMALES EXCEL REGULADOR CIRCUITO RLC .....	30
FIGURA A-1. SIMULACIÓN MODELSIM SOBREOSCILAMIENTO CIRCUITO RC .....	I
FIGURA A-2 SIMULACIÓN MODELSIM CIRCUITO RC .....	II
FIGURA A-3 SIMULACIÓN MODELSIM RLC .....	III

## ÍNDICE DE TABLAS

TABLA 3-1. EJEMPLO FORMATO COMA FIJA QX.Y.....	11
TABLA 4-1. EJEMPLO DE TRAMA.....	16
TABLA 4-2. TRAMAS DE EJEMPLO .....	16
TABLA 4-3. SELECCIÓN DE LOS BITS EN LAS TRAMAS.....	17
TABLA 5-1. COMPARACIÓN CIRCUITO RC.....	24
TABLA 5-2. COMPARACIÓN CIRCUITO RLC .....	28
TABLA 5-3. RECURSOS UTILIZADOS .....	30

# 1 Introducción

---

## 1.1 Motivación

Los reguladores digitales son dispositivos que están presentes en la vida cotidiana de las personas, ya sea en un termostato de calefacción o en la velocidad de crucero de un coche. Todos los dispositivos de control que años atrás trabajaban con señales analógicas se están modernizando y actualizando, utilizando señales digitales, las cuales tienen múltiples ventajas. Entre otras muchas, el control digital permite realizar reguladores más robustos, permite utilizar controles mucho más complejos y permite ser monitorizado y reconfigurado dinámicamente. El control digital es una disciplina que incluye, entre otros, conceptos matemáticos, aritméticos, además de conocer las dinámicas de la planta a controlar.

Además de los conceptos anteriormente enumerados, el control digital tiene una gran dependencia con el dispositivo que va a ejecutar dicho control. Esto es debido a que el dispositivo va a limitar en gran medida el tipo de control. Entre los diferentes dispositivos que se pueden utilizar para implementar reguladores de control están las FPGAs (*Field Programmable Gate Array*), microprocesadores, DSPs (*Digital Signal Processor*) y los PLC (*Programmable Logic Controller*). Las FPGAs permiten realizar lazos de control complejos y a alta velocidad gracias a las capacidades de concurrencia, pero como inconveniente destaca su complejidad de uso comparado con un procesador. Por todas estas razones, desarrollar un sistema de control real no es una tarea trivial.

En el grado de Ingeniería de Tecnologías y Servicios de Telecomunicación de la Universidad Autónoma de Madrid se imparte la asignatura Sistemas de Control. En las sesiones prácticas de la asignatura, los estudiantes deben tanto diseñar como probar reguladores digitales, tanto para un circuito RC (Resistencia-Condensador), y otro para un RLC (Resistencia-Bobina-Condensador). El diseño de los reguladores se realiza con la herramienta *Sisotool* de Matlab, lo que permite afianzar las competencias de la asignatura. Sin embargo, la implementación del regulador diseñado en una FPGA mediante VHDL supone una complicación añadida que, si bien los estudiantes están capacitados para ello, quita el foco de la asignatura en el control digital.

La principal complejidad de la implementación se debe al uso de la coma fija, más concretamente la librería *sfixed* de VHDL. Para poder optimizar tanto los recursos como maximizar la velocidad del regulador, se requiere que los reguladores trabajen con coma fija en vez de coma flotante. Sin embargo, la coma fija requiere que el diseñador calcule la posición de la coma de cada variable. Además, para poder realizar operaciones es habitual que las posiciones de la coma de los operandos deban ser alineadas.

Para contrarrestar esos problemas se sugieren varias alternativas como son la utilización de un microprocesador, la cual supondría una baja velocidad de funcionamiento, o el uso de DSP, con el que se produciría el mismo problema que con el microprocesador. Una solución fiable, pero costosa, es el uso de placas de control comerciales, no siendo una mala opción en cuanto a funcionamiento, pero el gran problema reside en el precio de estas.

En este TFG (Trabajo Fin de Grado) se propone el diseño e implementación de un regulador universal en coma fija, con capacidad de parametrización sin necesidad de

resintetizar el código de la FPGA. Es decir, el regulador podrá ser configurado mediante un ordenador y una interfaz gráfica, sin necesidad de que el estudiante deba implementar el regulador en la FPGA. De esa forma el estudiante podrá diseñar y probar numerosos reguladores, así como comparar los resultados experimentales con aquellos obtenidos mediante las simulaciones en Matlab. En definitiva, el esfuerzo de la asignatura durante las prácticas se centrará en los sistemas control, y no tanto en otros conceptos secundarios.

## 1.2 Objetivos

El objetivo principal de este Trabajo de Fin de Grado es la creación de un regulador digital universal capaz de ser implementado en una FPGA, a su vez dotado con un sistema de comunicación mediante una interfaz gráfica en la que el usuario pueda configurar los valores de dicho regulador sin modificar ni volver a resintetizar el código. Para lograr este objetivo se tienen que crear dos proyectos completamente distintos los cuales habrá que integrar al final.

- El primer proyecto es el correspondiente a la implementación del regulador digital utilizando coma fija parametrizable, para ello se implementará un controlador el cual es la parte más importante en todo sistema de control. Este dispositivo se encarga de conectar el regulador con el gestor ADC y PWM, en él se incluye un módulo VHDL que implementa un periférico UART.
- El segundo proyecto consiste en crear programa que configure el sistema de control mediante una interfaz gráfica, para ello se utilizará el lenguaje Java. Una vez se introduzcan los valores necesarios, el programa enviará todos los datos necesarios al controlador para que este los interprete y ponga en funcionamiento el regulador, todo ello transmitido mediante la UART con la biblioteca *SerialPort*.

Una vez ambos proyectos se han completado se obtendrá un sistema de control en lazo cerrado parametrizable por el usuario en tiempo real.

## 1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- En el segundo capítulo se explican los principales proyectos de innovación docente en el ámbito del control digital con el fin de que el alumno aprenda todos los conceptos relacionados con los sistemas de control por medio de pruebas experimentales
- El tercer capítulo detalla la forma de implementar el regulador digital universal, el diseño que va a tener y el tipo de coma utilizado, además de la adaptación del diseño a una FPGA.
- En el cuarto capítulo se muestra el diseño interior del sistema de control, componente a componente, desde la conexión con el ordenador, pasando por el gestor PWM, hasta llegar a la interfaz gráfica que permite configurar fácilmente el regulador en pocos segundos.
- En el quinto capítulo se explican los resultados obtenidos mediante las simulaciones obtenidas al conectar ambos proyectos al circuito y observar la salida obtenida por medio del osciloscopio o de otros programas. Además, se muestra una comparación entre la salida ideal del regulador y la simulada mediante un programa.
- Por último, en el sexto capítulo se muestran las conclusiones obtenidas de realizar este trabajo y los posibles futuros trabajos que puede tener.



## 2 Estado del arte

---

En este capítulo se recabará información sobre las distintas aplicaciones o proyectos que presentan plataformas orientadas al control digital o a la creación de laboratorios de docencia. Para explicar los diferentes proyectos existentes se procederá en primer lugar a introducir cómo funciona el proyecto creado en este trabajo.

El proyecto creado en este TFG está orientado a la docencia de los Sistemas de control. Una vez que un estudiante ha diseñado un regulador digital mediante herramientas como *Sisotool*, y una vez ha comprobado que el regulador teórico funciona correctamente, es especialmente importante que el estudiante pueda probarlo con un experimento real. Gracias a este proyecto, el estudiante introducirá dichas constantes en un formulario de una interfaz gráfica, y dichas constantes se mandarán a una FPGA que actúa como plataforma de control. Una vez se ha configurado el sistema el regulador comienza a funcionar y con esto el usuario podrá realizar múltiples experimentos reales sin tener que modificar el código implementado en la FPGA. Esto es importante puesto que el objetivo principal de dicha asignatura no es implementar código sino adquirir conocimientos sobre los sistemas de control. Una vez explicado el proyecto de elaboración propia será conveniente explicar los trabajos más relevantes en el ámbito de la docencia.

El primer proyecto que se procede a explicar está diseñado para docencia, centrado en la creación de un sistema de control orientado a aplicaciones mecatrónicas. En este proyecto se realiza el control en un ordenador mientras que la FPGA que contiene el sistema sirve de interfaz de sensado y actuación [1].

El proyecto descrito en [2] crea una aplicación 3D basada en web para crear un laboratorio de control. Se basa en la creación de una estación de trabajo remota, con una planta, un controlador digital con etapas de sensado y actuación además de admitir simulaciones a tiempo real. El estudiante puede manejar el laboratorio desde cualquier sitio mediante Internet. Los cálculos se realizan mediante ordenadores usando software avanzado como Matlab, se basa en el marco NCSLab 3D y se ha aplicado en la Universidad de Wuhan, China.

En [3] se muestra un laboratorio en tiempo real realizado mediante un ordenador, así como etapas AD y DA, creando programas de control basados en lenguaje C. Debido a la gran flexibilidad de los microprocesadores en acciones de control permite al estudiante una mayor versatilidad a la hora de experimentar con la aplicación.

Otro tipo de proyecto fue creado para ayudar a los estudiantes en la misma asignatura a la que va dedicada este trabajo, es decir, Sistemas de Control del grado de Ingeniería de Tecnologías y Servicios de Telecomunicación de la Universidad Autónoma de Madrid, en ella se creaba un laboratorio basado en web que permite a los estudiantes controlar de forma remota un puesto de laboratorio que consta de una FPGA, la planta a controlar, fuentes de alimentación y osciloscopios. El sistema permite subir ficheros para compilar y descargar en la FPGA. Sin embargo, los estudiantes tienen que codificar dichos ficheros HDL, por lo que se requieren de conocimientos previos de electrónica digital, aparte de los conocimientos de control [4].

En otros proyectos el laboratorio online está basado en Matlab, para compartir el equipo costoso se realiza un remoto mediante las interfaces TCP /IP o mediante la red de Internet/Intranet, todas las simulaciones se realizan a tiempo real [5].

Por otra parte, LabVIEW es un entorno de desarrollo integrado y multiplataforma para diseño de sistemas perteneciente a la empresa *National Instruments* [6]. Este programa es muy utilizado en los proyectos docentes donde el usuario es capaz de configurar el control de una planta mediante una interfaz gráfica. En [7-8] los alumnos pudieron crear y diseñar experimentos implementando el código muchas veces mediante herramientas gráficas. Otro dispositivo es myDAQ, creado por *National Instruments* con un precio para docencia de 350€ que permite la adquisición de datos, así como la actuación. Si el estudiante quiere realizar sistemas de control deberá utilizar las herramientas LabVIEW que se ejecutan en un ordenador [11]. Otro dispositivo de la misma empresa es myRIO. Este dispositivo es de mayor coste, lo que conlleva mayores prestaciones puesto que contiene una FPGA que puede ser programada para adaptarse al control necesario. En este caso, el control puede realizarse mediante LabVIEW, o se puede descargar a la FPGA. La programación de la FPGA se puede realizar de forma gráfica, pero las opciones en muchos casos están muy limitadas [12-13].

En la Universidad de Alcalá de Henares se realizó el control remoto de un actuador de movimiento basado en un motor de corriente continua. La conexión se realiza mediante una tarjeta de evaluación miniDK2, basada en un procesador NXP LPC1768 con núcleo ARM. Se plantea el diseño de un control PID para controlar la velocidad de un motor DC en tiempo continuo, otro diseño es el de un control PID para manejar un sistema de motor LEGO haciendo uso de *Simulink* que es otra herramienta de Matlab [9].

No todos los proyectos creados son docentes, ni gratuitos, por ejemplo, la SpCard de un sistema de control basado en FPGA de altas prestaciones con PWMs de 25 MHz y 10 canales con ADC de 1 MSP. Pero la desventaja que producen estos dispositivos es su precio, en este caso es superior a los 1000€ [10].

## 3 Diseño

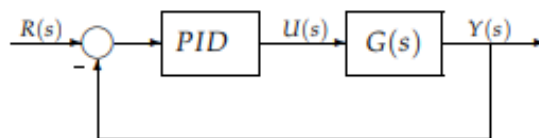
### 3.1 Reguladores PID

Para explicar el diseño del presente proyecto, será necesario estudiar con anterioridad los sistemas de control para comprender mejor lo que es un regulador. Por sistema de control se entiende cualquier dispositivo encargado de regular o dirigir otro sistema [16]. El ejemplo por excelencia de cualquier controlador o sistema de control es el termostato de la calefacción, cuyo mecanismo permite marcar la temperatura a la que el usuario desea estar. El control que realiza este aparato para conseguir dicha finalidad se basa en elevar la temperatura hasta alcanzar la deseada, y mantenerla constante durante el tiempo que sea necesario.

Una vez se ha explicado el propósito de los sistemas de control, se procederá a explicar los reguladores, así como sus tipos más importantes. Estos sistemas pueden ser dividirse en dos subgrupos en función del tipo de lazo que presentan. La diferencia entre ambos recae en si el sistema tiene realimentación o no. El primer tipo es el de lazo abierto, mientras que el otro subgrupo es el de lazo cerrado. El lazo abierto no tiene realimentación a diferencia del lazo cerrado.

En el diseño del presente trabajo ha sido elegido como elemento de control un regulador PID, que, como dictan sus siglas, realiza tres tipos de trabajos de control: proporcional, integral y derivativa. Estos tres tipos de trabajos varían dependiendo del control que realice, pues la fórmula a la que es sometida la señal cambia en función de los tipos de este.

Por ello, en primer lugar, se incluye el regulador en el diagrama de bloques mostrado anteriormente. Para ello se incluye primero el regulador en el diagrama de bloques nombrado anteriormente. En la figura 3-1 se muestra un ejemplo gráfico de un sistema de control utilizando como regulador uno del tipo PID.



**Figura 3-1. Sistema de control con PID**

El primer bloque que nos encontramos es el regulador PID que se explicará a continuación, después de él se encuentra el proceso, al que en el dibujo llama  $G(s)$ . Para explicar el regulador PID se nombran las señales del sistema de la misma manera que en la figura 3-1.  $R(s)$  corresponde a la señal de entrada e  $Y(s)$  a la señal de salida, se presentan las operaciones que se realizan en el regulador, donde la señal  $e(t)$  se corresponde con la intermedia entre el restados y el PID, la cual recibe el nombre de error. Por tanto, procedemos a exponer las distintas acciones de control que puede realizar este regulador:

- Acción de control proporcional:  $u(t) = K_p * e(t)$ , donde  $K_p$  es una ganancia ajustable a cada diseño.
- Acción de control que se realiza es control integral:

$$u(t) = \int_0^t e(\tau) d\tau$$

**Figura 3-2. Acción de control proporcional**

- Si la acción de control es proporcional-integral (PI), la ecuación de salida se define de la siguiente manera:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau$$

**Figura 3-3. Acción de control proporcional-integral**

Donde se observa que es una combinación de las dos técnicas anteriormente explicadas. Esta técnica se adecua al escalón digital, puesto que va aumentando progresivamente hasta llegar al máximo.

- Acción de control proporcional-derivativa, está contiene la derivada temporal del error.

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

**Figura 3-4. Acción de control proporcional-derivativa**

- Acción de control proporcional-integral-derivativa:

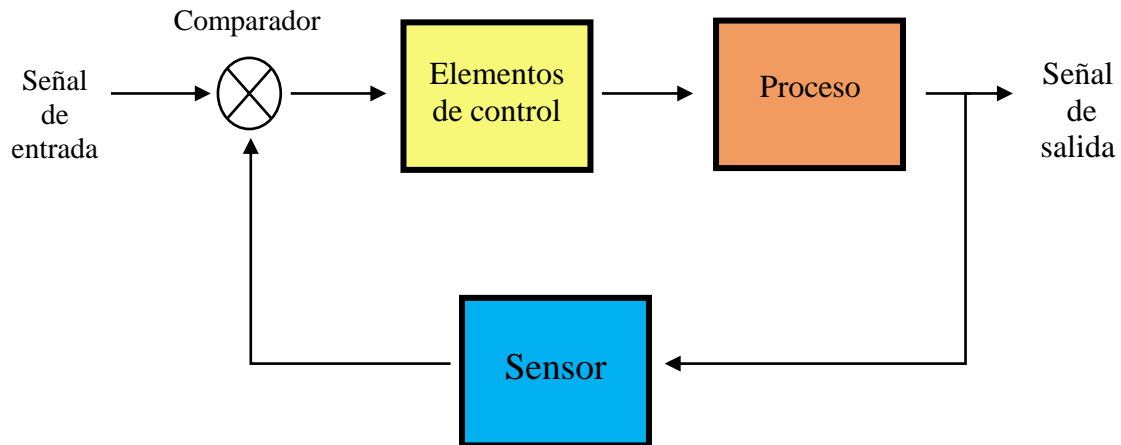
$$K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt}$$

**Figura 3-5. Acción de control proporcional-integral-derivativa**

Todas estas son las opciones que nos ofrece el regulador PID, cada una de ellas depende del diseño y resultados que se desean obtener [15].

### **3.2 Lazo cerrado**

Como se ha citado anteriormente, el diseño creado para el regulador universal va a estar implementado mediante un sistema en lazo cerrado, como se puede observar en la figura 3-6. El error de un sistema en lazo cerrado se define como la diferencia entre la señal de entrada y la realimentación proveniente del sensor.



**Figura 3-6. Sistema de control de lazo cerrado**

El diseño creado será distinto al que podemos observar en la figura puesto que existe una señal de referencia la cual da el valor al que el usuario quiere llegar una vez se ponga en funcionamiento el regulador. En cuanto a los elementos que están contenidos en el sistema el comparador se encarga como bien dice su nombre de la comparación entre la señal de entrada y la señal proveniente de la salida del sensor, el sensor no es necesario que contenga valores, pudiendo ser de valor uno. Si el valor del sensor es nulo quiere decir que el sensor no deja pasar ningún tipo de señal y, por tanto, no habría realimentación, lo que lo convertiría en un sistema de lazo abierto.

En cuanto a los elementos de control se pueden utilizar distintos tipos de reguladores, en este diseño se utilizará el explicado anteriormente, regulador PID. Una vez la señal atraviesa este elemento se dirige hacia el bloque denominado proceso, aunque también puede ser llamado planta. Este elemento se encarga de realizar operaciones determinadas, en este caso, se encarga de contener los elementos necesarios para regular el sistema como pueden ser condensadores o bobinas. Por último, el sensor se encarga de traducir la información que proviene de la salida y convertirla en impulsos digitales.

Puesto que la parte más importante de todo el sistema de control recae en el regulador, es la parte que más importancia se le va a dar. Como se ha dicho anteriormente se pretende crear un regulador universal capaz de controlar cualquier señal de entrada, por tanto, se adjunta una fórmula correspondiente a un regulador universal, esta fórmula será explicada más adelante mediante las ecuaciones en diferencias.

$$R(z) = \frac{A_0 + A_1 * z^{-1} + A_2 * z^{-2} + A_3 * z^{-3}}{1 - B_1 * z^{-1} - B_2 * z^{-2} - B_3 * z^{-3}}$$

**Figura 3-7. Fórmula de un regulador universal**

### 3.2.1 Ecuaciones en diferencias

Para poder explicar correctamente la fórmula anterior hay que explicar que son las ecuaciones en diferencia. Una función básica en diferencia es del tipo  $f(n+1)$ ; esto significa que selecciona el valor de la función, pero en un tiempo anterior, concretamente un valor

antes. Para poner un ejemplo más práctico se procede a descomponer la ecuación anterior a otra en diferencias.

$$G(z) = \frac{A_0 + A_1 * z^{-1} + A_2 * z^{-2} + A_3 * z^{-3}}{1 - B_1 * z^{-1} - B_2 * z^{-2} - B_3 * z^{-3}} = \frac{Y(z)}{X(z)}$$

Donde  $Y(z)$  se corresponde con la salida y  $X(z)$  se corresponde con la entrada. Si se desarrolla la ecuación anterior obtenemos la ecuación en diferencias.

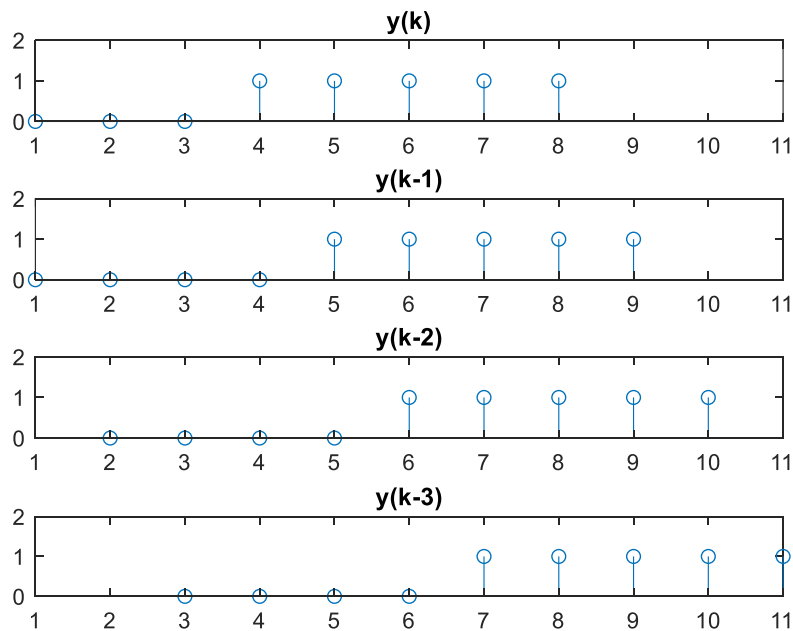
$$X(z) * (A_0 + A_1 * z^{-1} + A_2 * z^{-2} + A_3 * z^{-3}) = Y(z) * (1 - B_1 * z^{-1} - B_2 * z^{-2} - B_3 * z^{-3})$$

Si despejamos la  $Y(z)$  se obtiene la salida del sistema:

$Y(z) = -B_1 * Y(z^{-1}) - B_2 * Y(z^{-2}) - B_3 * Y(z^{-3}) + A_0 * X(z) + A_1 * X(z^{-1}) + A_2 * X(z^{-2}) + A_3 * X(z^{-3})$  Una vez se tiene la ecuación separada, se procede a ponerla en forma diferencias, esto equivale a sustituir las “z” por la función en diferencias. Para obtener la función que diseña el regulador se divide la salida entre la entrada, a esto se le denomina función de transferencia del regulador.

$$y(k) = -B_1 * y(k-1) - B_2 * y(k-2) - B_3 * y(k-3) + A_0 * x(k) + A_1 * x(k-1) + A_2 * x(k-2) + A_3 * x(k-3)$$

Se puede observar que las variables con grado menor o igual a 1 sufren un desplazamiento en la ecuación en diferencias, cuanto menor es el grado mayor es su desplazamiento. Un ejemplo más sencillo se puede realizar desplazando la señal escalón como se muestra en la siguiente figura:



**Figura 3-8. Ejemplo desplazamiento señal**

Como se puede comprobar en esta figura, cuanto mayor es el desplazamiento que se realiza a una señal, más tarde comienza a tener el valor asignado, en este caso 1. Por ejemplo, en el último caso tarda 3 ciclos más en obtener el primer 1 en comparación con la señal sin desplazar.

### 3.2.2 Diseño del regulador

Una vez se ha propuesto la fórmula del regulador es la hora de proponer un diseño software el cual sea capaz de realizar dicha ecuación. Por ello se introducen una serie de Flip Flops para que retrasen la señal en un ciclo, siendo el equivalente a los exponentes de las  $z$  en la fórmula. Por tanto, habrá tantos Flip Flops como desplazamientos necesite la señal. Además, se incluyen una serie de multiplicadores para realizar el escalado a cada estado con el fin de conseguir los resultados obtenidos. Todos estos valores serán introducidos por el usuario durante la ejecución del programa. La señal del ADC es un valor el cual se asigna fuera del regulador, este valor se le resta a la referencia para comprobar si el regulador tiene que aumentar o disminuir la cantidad pedida.

Todos los Flip Flops se conectan mediante una señal llamada “Enable” la cual hace que los componentes se activen a la vez produciendo así que los datos consigan atravesar el dispositivo a la vez, normalmente este se pone a ‘1’ a la vez que el reloj del sistema.

A continuación, se muestra el diseño realizado en este proyecto:

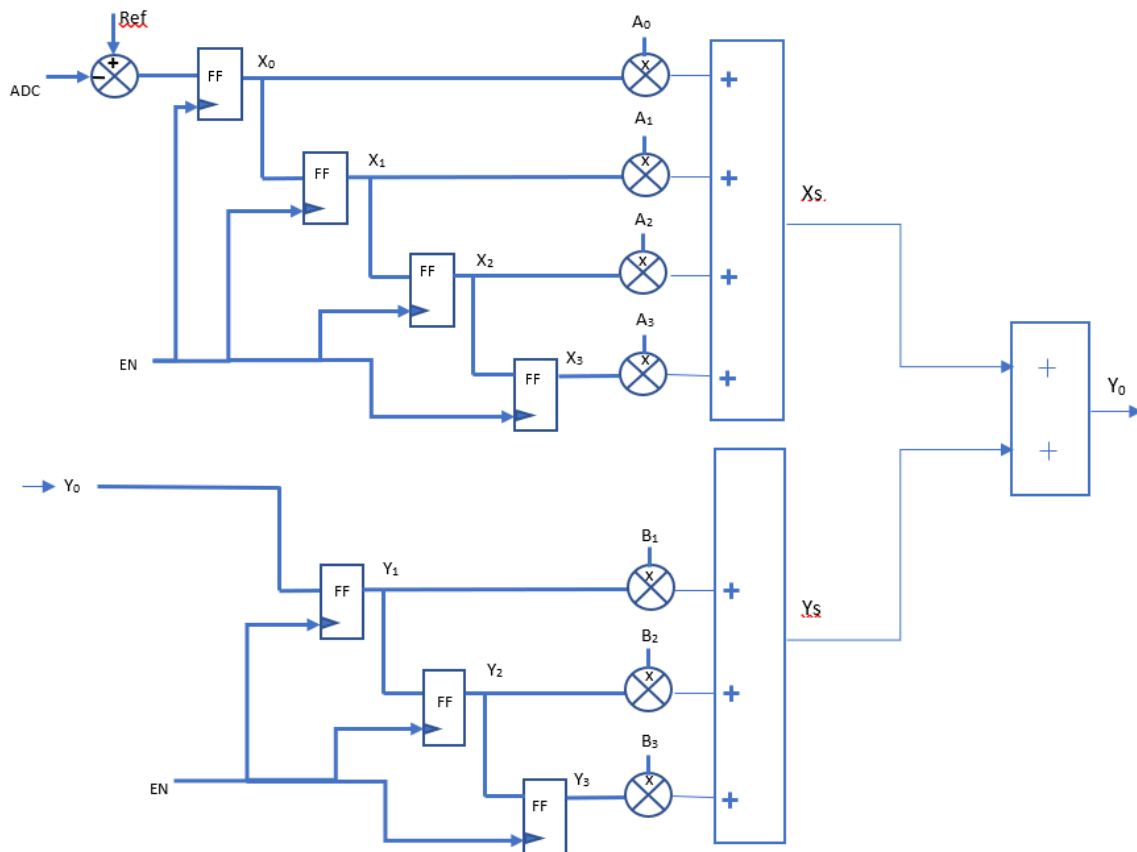


Figura 3-9. Diseño de un regulador PID en VHDL

Se puede observar que en el diseño hay una señal de referencia dicha anteriormente, además de unos multiplicadores, estos contendrán los valores para conseguir que el regulador sea óptimo. Una vez se han obtenido las multiplicaciones se realiza una suma de todos los operandos calculados anteriormente, con ello se obtiene tanto el numerador, mediante la suma de todas las  $X$ , como del denominador, que se alcanza mediante la suma

de todos los componentes  $Y$ . Con estos valores se consigue la función de transferencia del regulador nombrado anteriormente.

Como se puede ver en la ecuación en diferencias, la salida del regulador depende de las salidas anteriores. Por tanto, la salida del regulador es realimentada para calcular los términos  $Y_1$ ,  $Y_2$ , e  $Y_3$ , y con ellos obtener de nuevo la salida. Mediante la creación de un regulador universal se puede controlar cualquier tipo de circuito, en nuestro caso se probará sobre un circuito resistencia-condensador (RC) y otro resistencia-bobina-condensador (RLC).

### **3.3 Adaptación a la FPGA**

Debido a la necesidad de conseguir un regulador que funcione rápido, la mejor posibilidad es el uso de la adaptación del circuito a la FPGA, para ello todos los multiplicadores deben ser DSPs de 17x17 bits, puesto que todas nuestras señales son de 17 bits a excepción de los desplazadores que son de 6 bits. Se han elegido 17 bits de tamaño de bus puesto que la familia Spartan 3, placa sobre la cual se trabaja tiene DSPs de 17x17 bits [17], así se consigue una mayor rapidez en los cálculos, si necesitásemos una longitud de bits mayor no habría problemas de funcionamiento, pero se necesitará el uso adicional de LUTs para completar el cálculo de los bits añadidos.

Para conectar el circuito a la FPGA se realiza mediante el uso de los pines conectores que contiene la placa, al haber tres pines se elige uno de ellos. Para realizar correctamente la conexión se utilizará un adaptador que conectará a los pines el circuito que se desea regular. Los valores correspondientes a las constantes de los multiplicadores se enviarán a la placa mediante un conector tipo DB9 y un protocolo serie.

La FPGA funciona a una frecuencia de 50 MHz, en cambio, el regulador creado se ha diseñado para que funcione el doble de rápido, es decir, a 100 MHz. Esto se consigue mediante la introducción de un DCM (digital clock manager), mediante este componente se duplica la frecuencia de reloj, consiguiendo así los 100 MHz deseados, esto provoca una mayor velocidad de uso del regulador.

Visto el diseño esquemático en la figura 3-9, se debe elegir el tipo de aritmética que se va a utilizar. Se abren dos posibilidades, coma fija o coma flotante. La elección más fácil es la coma flotante puesto que permite de forma nativa tener decimales y el diseñador no debe preocuparse del lugar de la coma, puesto que la aritmética la resuelve de forma automática. En cambio, tiene una gran desventaja que es la velocidad, puesto que en muchos casos puede llegar a ser hasta 10 veces más lenta que la coma fija [18]. Otro problema es la gran cantidad de recursos utilizada en comparación con la otra opción. La otra opción es la coma fija, esta posibilidad optimiza la velocidad y recursos. Sin embargo, el diseñador debe preocuparse del lugar que ocupa la coma en cada variable antes de realizar operaciones aritméticas, lo cual resulta tedioso. Para solucionar todos estos problemas, se propone el uso de la coma fija parametrizable que combina ambas propuestas.

### **3.4 Coma fija parametrizable**

Se quiere diseñar un regulador cuyos coeficientes, entradas y salidas sean totalmente parametrizables. Esto quiere decir que el diseño tiene que poder soportar valores muy



dispares dependiendo de la aplicación. Sin embargo, se pretende que el sistema sea parametrizable en tiempo real, no en tiempo de síntesis. Esto provoca que la posición de la coma de cada variable debe poder ser modificada sin tener que volver a sintetizar. Por ello no se podrán utilizar bibliotecas de coma fija como *sfixed*.

Se utiliza el formato de coma fija QX.Y. Este diseño tiene X bits de parte entera e Y bits de parte decimal además de un bit adicional correspondiente al signo, pudiendo declarar tanto números positivos como negativos. Para explicar mejor cómo funciona la coma fija se realiza mediante un ejemplo. Si se tiene un formato Q2.3, tenemos 2 bits de parte entera y 3 de parte decimal, además de un bit adicional de signo, por tanto, el número máximo positivo y mínimo negativo sería:

Bit	2	1	0	,	-1	-2	-3
Peso	$2^2$	$2^1$	$2^0$		$2^{-1}$	$2^{-2}$	$2^{-3}$
Valor máximo positivo en formato Q2.3 (3,875)	0	1	1		1	1	1
Valor máximo negativo en formato Q2.3 (-4)	1	0	0		0	0	0

**Tabla 3-1. Ejemplo formato coma fija QX.Y**

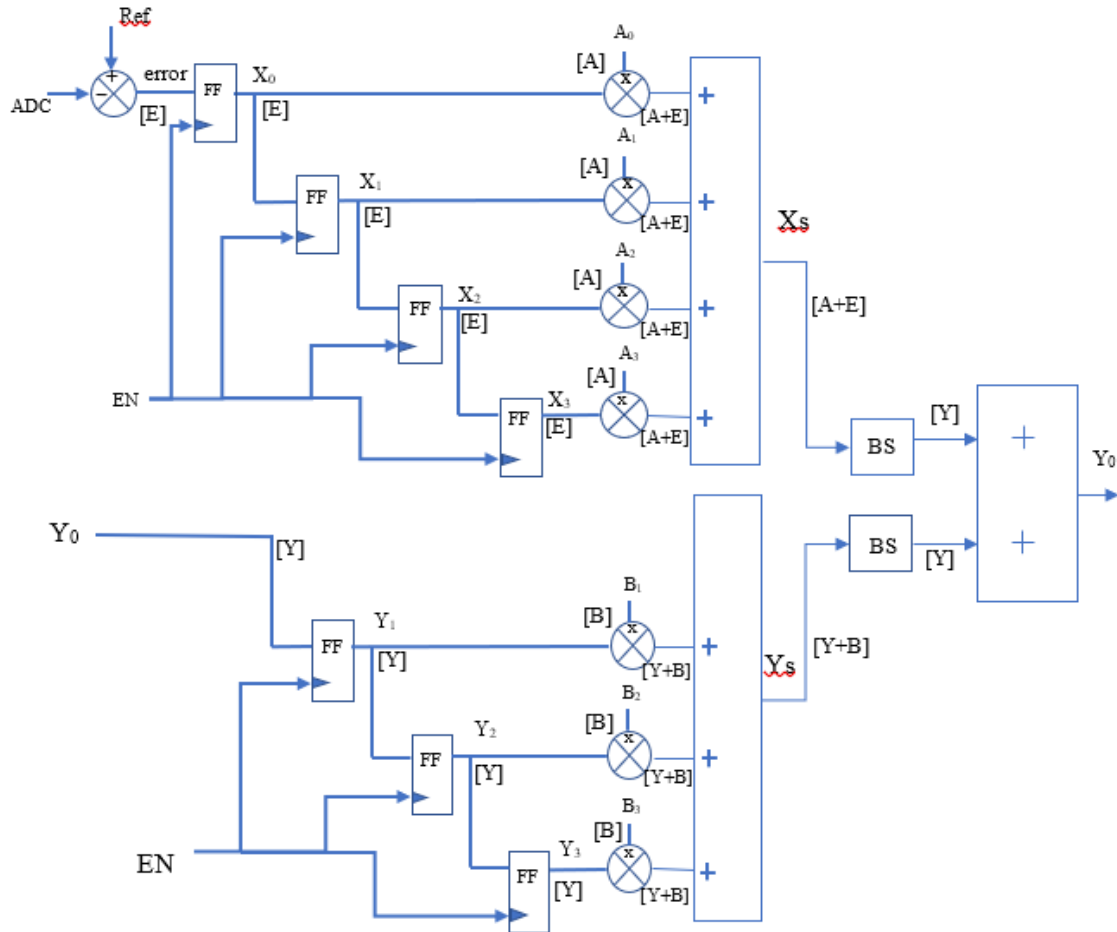
Donde el 0 equivale al bit de signo, puesto que 0 significa que será un número positivo y 1 corresponde al número negativo, a continuación, tenemos dos bits de parte entera y después los tres bits de parte decimal, por tanto, el mayor número positivo posible en ese formato es 3,875. A diferencia del número positivo en este el bit de signo vale 1 como se ha comentado anteriormente. Por tanto, si se transforma el número a decimal, este valdría -4.

Como se ha expuesto anteriormente el diseño del regulador únicamente en coma fija exige una mayor dificultad a la hora de realizar el código que implemente el regulador, por ello se propone un diseño que combina las ventajas de cada formato, la rapidez de la coma fija y la adaptabilidad de la coma flotante. Esta combinación se denomina coma fija parametrizable la cual utiliza menos recursos a una mayor velocidad, además de utilizar la facilidad de la coma flotante para adaptarse a los cambios de valores. Pero al realizar este diseño surge un problema, puesto que dos números en coma fija deben tener el mismo número de decimales para poder ser tanto sumados como restados. La alineación de la coma se puede realizar fácilmente mediante desplazamientos aritméticos. Sin embargo, como en un principio no se sabe la posición de la coma de una variable, ya que estos números dependen de la aplicación del usuario, los desplazamientos deben ser variables. Para ello introducimos un dispositivo llamado *barrel shifter*. Estos dispositivos se van a introducir después de la suma de todas las X's y las Y's, con el fin de conseguir que todo esté en la misma escala. La escala es el número de decimales que tiene la variable sobre la que se está trabajando y se calcula de la siguiente manera:

$$Escala = Tamaño_{señal} - Máx_{superior}(1 + \log_2(\max(A_0, A_1, A_2, A_3)))$$

Las demás escalas se calcularían igual, pero con la diferencia de las variables introducidas, el caso anterior se corresponde con la escala A. En la escala B se seleccionan las variables B<sub>1-3</sub>. Solo se usarán las variables con valores fijos para calcular dichas escalas, no se usarán los valores que cambian a lo largo de la ejecución. Por tanto, para calcular la escala Y se usará un valor de salida máximo fijo de 3,3 voltios.

El desplazamiento en cambio se calcula mediante la resta de las escalas final e inicial de cada operando. Por tanto, el esquema del regulador va a sufrir una serie de variaciones con el fin de introducir esos *barrel shifter*, cuyo nombre abreviado en el dibujo será BS, además introduciremos los nombres de las escalas entre corchetes.



**Figura 3-10. Diseño del regulador completo**

Como se ha citado anteriormente, se introducen los desplazadores justo después de los sumatorios tanto de las X's como de las Y's, en el primer caso el valor del desplazamiento sería  $Y-(A+B)$  en valor absoluto, mientras que en el segundo caso valdría  $Y-(Y+B)$ . Así se consigue que el número se encuentre en la escala deseada. El número resultante se transforma de decimal a binario, para obtener cuantos bits se ha de desplazar la coma, pero todavía falta saber si se desplazará hacia la izquierda o la derecha. Para averiguar el sentido del desplazamiento hay que mirar que operando es mayor, en nuestro caso el operando entrante es  $(A+B)$ , mientras que el operando de salida es  $Y$ . Si el operando entrante es mayor al de salida el sentido es 0 y por tanto la coma se mueve hacia la derecha, en cambio, si la salida es mayor su valor será 1 y el desplazamiento de la coma será hacia la izquierda.

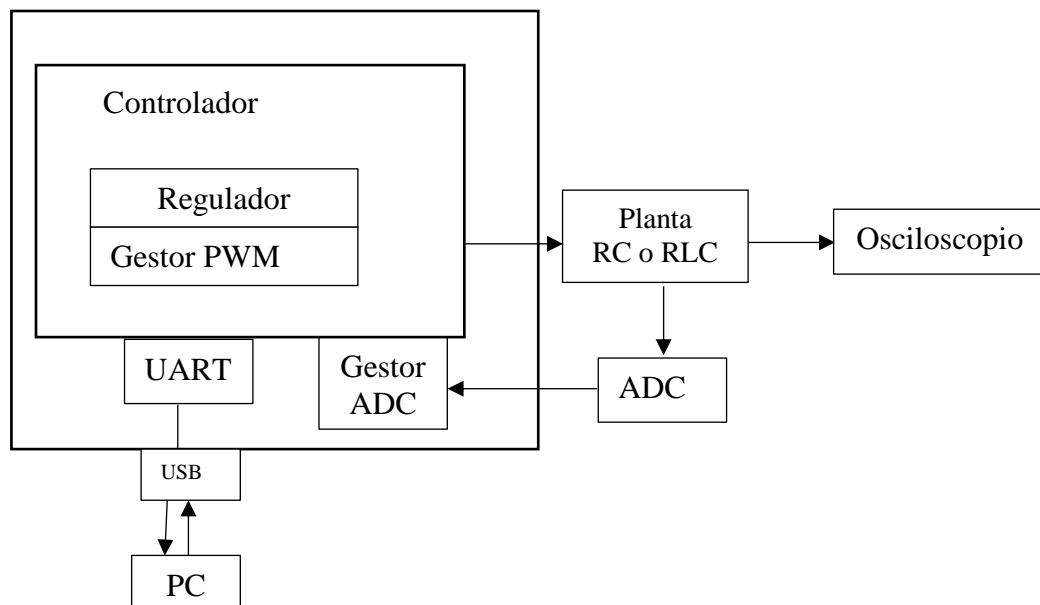
Una vez explicado el formato de las variables que se van a utilizar, ya se puede diseñar por completo el regulador. Para que este no sature se implementa un bloque de saturación después de la suma de  $X_s$  e  $Y_s$ . Tanto el valor máximo como mínimo de la salida tienen bloques de saturación que evita desbordamientos. Además, gracias a la aplicación el

usuario podrá poner límites aún más restrictivos para evitar que el regulador sobrepase los umbrales deseados por el usuario.



## 4 Integración del controlador

Para lograr una implementación que permita diseñar la coma fija parametrizable será necesario incluir un módulo de comunicación entre el usuario y la FPGA. Para ello se creará un formulario en el entorno de programación Java que permita al usuario introducir los valores de las variables en formato decimal. Los datos se enviarán a la FPGA mediante un conversor USB-UART (*Universal Asynchronous Receiver-Transmitter*). La FPGA, por su parte, tiene implementado un módulo intérprete para recibir las tramas de configuración recibidas.



**Figura 4-1. Diseño del sistema completo**

En la figura 4-1 se puede observar el diseño del sistema completo. La parte más importante de todo este sistema es el controlador, el encargado de que todos los demás componentes funcionen correctamente y trabajen al unísono. La comunicación entre el controlador y el ordenador como se ha citado anteriormente se realiza mediante el periférico UART, el cual utiliza el conector tipo DB9 con un protocolo en serie. Después del controlador se encuentra la planta con posibles circuitos como RC o RLC entre otros. Ese circuito devuelve una salida de voltaje digital la cual es interpretada mediante un conversor digital-analógico que nos devuelve señales analógicas listas para interpretar mediante el osciloscopio.

### 4.1 Controlador

El bloque correspondiente al controlador es el encargado de conseguir que todos los demás bloques trabajen al mismo tiempo. Aparte de realizar esta función, también es el encargado de que todas las señales entrantes y salientes de los bloques estén conectadas entre sí en el lugar correcto, así como de mantener la conexión con el PC. Esta conexión se realiza mediante el dispositivo UART.

### 4.1.1 UART

El regulador debe ser configurado con todas las constantes dichas en la fórmula y los cambios de escala correspondientes, para ello se ha desarrollado un protocolo de comunicación basado en tramas UART. En la parte del ordenador, se utiliza un conversor USB-UART y se utilizan las funciones JAVA de comunicación con puertos UART. En la parte de la FPGA se ha utilizado el módulo VHDL que implementa un periférico UART. Este módulo se ha extraído del procesador de bajos recursos *picoBlaze* de Xilinx

Esto se realiza mediante el envío de tramas, los cuales serán diferentes para cada variable. La conexión se va a ejecutar con una velocidad de 115200 bits/s, obteniendo un tiempo entre datos de alrededor de 8,68 microsegundos. Esta velocidad es la máxima del conversor USB-UART utilizado.

Este método envía tramas de byte en byte, por ello, tiene un buffer en el cual se va almacenando el dato hasta que este está completo. En este diseño ha sido necesario el uso de cinco bytes para poder mandar el valor completo de la variable correspondiente. Esto es debido a que es necesario mandar una cabecera para definir dónde se debe escribir el dato que se envía, es decir, a qué constante del regulador debe ir dicho valor. Además, se ha añadido una suma de control o checksum para comprobar la integridad de la trama. Por ello en la implementación hay una señal que indica cuándo el dato está completo para poder leerlo. La mayoría de las señales utilizadas son de 17 bits a excepción de los desplazadores, por ello se mostrará un ejemplo tanto de una como de otra. El primer paso consiste en transformar el dato a la escala correspondiente, multiplicando el dato por  $2^{\text{escala}}$ . Una vez se ha realizado ese paso se convierte el número a binario.

	Cabecera		Dato		Dato		Dato		Checksum	
Bits	39	32	31	24	23	16	15	8	7	0

**Tabla 4-1. Ejemplo de trama**

Visto el ejemplo de la estructura de la trama en la tabla 4-1, se procede a explicar un ejemplo real en una señal de 17 bits y un desplazador los cuales necesitan 7 bits, 6 de desplazamiento y uno de sentido. Para la señal de 17 bits, se enviará el número 1,2 y será el primer número que se envíe. Este número tiene una escala de valor 14, por tanto, el número a enviar es de 19660. Además, se supone que el identificador (cabecera) de comando será el 0x01. Para el ejemplo del desplazador, el valor del desplazamiento es de 16, con sentido 1, además se enviará en la décima posición. En este caso no hay que aplicar ninguna escala, puesto que los desplazadores se configuran con números enteros. En este segundo ejemplo se considerará que la cabecera de comando será 0x0A.

Dato para enviar: 19660	Valor en binario: 00100110011001100	Dato para enviar: desplazamiento 16, sentido 0	Valor en binario: 010000 1
Primer byte	0000 0001 ( <b>0x01</b> )	Primer byte	0000 1010 ( <b>0x0A</b> )
Segundo byte	0010 0110 ( <b>0x26</b> )	Segundo byte	1000 0000 ( <b>0x80</b> )
Tercer byte	0110 0110 ( <b>0x66</b> )	Tercer byte	0100 0000 ( <b>0x40</b> )
Cuarto byte	0000 0000 ( <b>0x00</b> )	Cuarto byte	0000 0000 ( <b>0x00</b> )
Quinto byte	<b>01+26+66+00 = 8D</b>	Quinto byte	<b>0A+80+40+00 = CA</b>

**Tabla 4-2. Tramas de ejemplo**

En la tabla 4-2 los valores que se encuentran en negrita se corresponden con su valor en hexadecimal. Además es posible observar cómo en el ejemplo de 17 bits, en el cuarto byte, se rellena con ceros para así completar el byte. En cambio, en el desplazador se rellenan los dos últimos bits del tercer byte y también el cuarto byte completo. En el primer byte el sentido del desplazamiento iría situado en el bit más significativo y los demás serían cero. En la tabla 4-3 se muestra los bits seleccionados una vez la trama ha sido recibida.

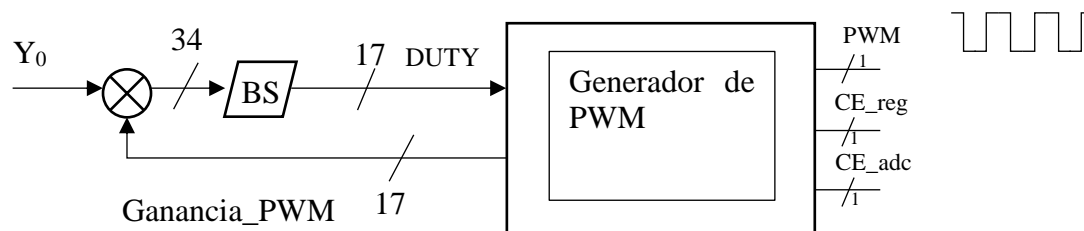
Trama	Primer byte	Segundo byte	Tercer byte	Cuarto byte	Quinto byte
Señal 17 bits	0000 0001	0010 0110	0110 0110	0000 0000	1000 1101
Desplazador	0000 1010	1000 0000	0100 0000	0000 0000	1100 1010
Bits	39 32	31 24	23 16	15 8	7 0

**Tabla 4-3. Selección de los bits en las tramas**

En color verde se muestran los bits seleccionados para realizar los cálculos, mientras que los de color rojo no se utilizan dentro del controlador.

### 4.1.2 Gestor PWM

El gestor PWM es uno de los bloques más importantes del controlador. Será el encargado de gestionar el valor de la señal de PWM, cuya señal es la salida que actuará sobre la planta, el regulador genera una consigna de PWM. Con esta consigna se genera un PWM, el PWM es la actuación sobre la planta. Por tanto, el objetivo del gestor es controlar la tensión media de salida tras un circuito que actúa como filtro RC o RLC. Debido a esto, cuando la señal está a '1' aumentará la tensión de salida, en cambio, cuando esté a '0' la tensión disminuirá. Este gestor sigue el siguiente diseño:



**Figura 4-2. Diseño del gestor PWM**

Como puede observarse en la figura 4-2, el gestor PWM no genera únicamente el PWM, sino que también genera los pulsos de tipo *chip enable* para sincronizar tanto el regulador como el gestor ADC con la actuación del sistema. Internamente el gestor tiene una función rampa que va incrementándose en uno ciclo cada ciclo. Inicialmente la salida estará a 1, pero cuando la rampa llega al valor de  $Y_0$  multiplicado por la ganancia de la figura, la salida cambiará a 0. El valor *DUTY* de la figura 4-2 es el valor obtenido de la multiplicación de la salida del regulador con la ganancia del PWM y desplazada tantos bits como sea necesario.

Resumiendo lo anteriormente expuesto y de modo aclaratorio, el gestor PWM por lo tanto, será el encargado de generar la actuación del sistema mediante una señal PWM.

Otros componentes a destacar, y no menos importantes que los anteriores, serían el regulador y el gestor ADC. Por otro lado, el dispositivo ADC sería el encargado de gestionar

la comunicación con un ADC comercial (*ADC7813*) que servirá para mirar la salida de la planta y poder realizar control en lazo cerrado. Como se ha comentado anteriormente, existe sincronización entre la actuación y el sensado mediante el ADC, pudiendo el usuario configurar dicha sincronización. Esta característica es interesante en los sistemas de control, especialmente si la fecha sensada presenta mucho rizado.

## **4.2 Interfaz gráfica**

Se ha diseñado un programa que es capaz de configurar mediante una interfaz gráfica el sistema de control, el lenguaje utilizado para realizar ese programa es Java puesto que es un lenguaje de alto nivel y multiplataforma, lo que facilita el futuro traslado a otros sistemas operativos (actualmente Windows). Además, para facilitar el diseño de la interfaz se ha usado el Framework JavaFX.

En ella se introducen todas las fórmulas encargadas de la creación de todas las variables para que el usuario pueda cambiar los valores del regulador fácilmente. Por ejemplo, se introducen las fórmulas de las escalas y la obtención de los valores de desplazamiento. El objetivo de esta interfaz es que el usuario pueda observar la variación del voltaje de salida únicamente cambiando las variables del regulador y la referencia sin tener que introducirse en el código VHDL sobre el que está hecho el controlador. Gracias a la arquitectura elegida, no es necesario resintetizar el código de la FPGA para implementar un nuevo regulador, sino que simplemente éste debe ser reconfigurado mediante las nuevas constantes a través de la aplicación gráfica. Dicha aplicación se encarga del cálculo de todas las escalas (número de decimales), y de la configuración de los desplazadores (*barrel shifters*). Por tanto, el usuario no tiene que conocer cómo está hecho internamente el sistema de control.

La comunicación se ha implementado mediante un hilo diferente para cada variable para así no bloquear la interfaz gráfica. Para la comunicación se ha utilizado la biblioteca UART llamada *SerialPort*. En la figura 4-3 se observa el diseño gráfico del formulario, así como un ejemplo de regulador para una planta RLC. Hay valores incluidos por defecto que están puestos de acuerdo con las limitaciones del sistema, como, por ejemplo, que la salida máxima sea de 3,3 voltios.



**Laboratorio de Sistemas de Control**  
**EPS - UAM**

**Configuración del regulador**

A0	<input type="text" value="0.25"/>
A1	<input type="text" value="-0.4375"/>
A2	<input type="text" value="0.19140625"/>
A3	<input type="text" value="0"/>
B0	<input type="text" value="1"/>
B1	<input type="text" value="1"/>
B2	<input type="text" value="0"/>
B3	<input type="text" value="0"/>

$$R(z) = \frac{A_0 + A_1 * z^{-1} + A_2 * z^{-2} + A_3 * z^{-3}}{1 - B_1 * z^{-1} - B_2 * z^{-2} - B_3 * z^{-3}}$$

**Configuración del sistema**

Máximo valor de entrada (V)	<input type="text" value="3.3"/>
Valor máximo Vout (V)	<input type="text" value="3.3"/>
Valor máximo saturación (V)	<input type="text" value="0.95"/>
Valor mínimo saturación (V)	<input type="text" value="0"/>
Ganancia ADC	<input type="text" value="0.012941176"/>

**Conexión a la FPGA**

▶ Opciones avanzadas

Valor de Referencia (V)	<input type="text" value="1.65"/>	Valor máximo de voltaje al que quieres que llegue la simulación
-------------------------	-----------------------------------	---

**Figura 4-3. Formulario de las variables**

Una vez se ha rellenado el formulario entero, se puede conectar la placa FPGA, y con ello se enviarán todas las variables al controlador con el fin de obtener la salida que se observará en el osciloscopio. Para dar facilidad al usuario y poder comprobar como varía el regulador en función de la referencia, el formulario consta de un botón con el que puede actualizarse únicamente la referencia y que el regulador siga funcionando.



## 5 Integración, pruebas y resultados

Una vez se han implementado todos los bloques, es la hora de realizar una serie de pruebas con el fin de comprobar si el diseño se ha realizado correctamente. Para ello se va a probar el sistema diseñado con dos reguladores distintos.

Para el diseño de los reguladores se ha utilizado la herramienta *Sisotool* de Matlab [14]. Esta herramienta permite, una vez indicada la planta a regulador, diseñar un controlador observando la respuesta en lazo cerrado del sistema completo.

Las plantas seleccionadas como ejemplo son un circuito RC, y otro circuito RLC. Para crear el fichero de Matlab primero hay que definir la función de transferencia del circuito (planta del sistema).

### 5.1 Circuito RC

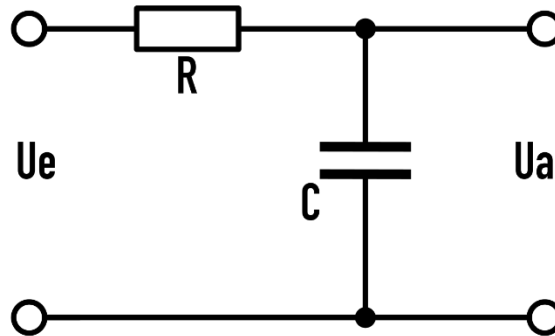


Figura 5-1. Circuito RC

$$G(w) = \frac{V_a}{V_e} = \frac{Z_c}{R + Z_c} \rightarrow G(w) = \frac{\frac{1}{jwC}}{R + \frac{1}{jwC}} = \frac{1}{1 + jwRC} \rightarrow G(z) = \frac{1}{1 + zRC}$$

Donde los valores de  $R=47K\Omega$  y  $C=22nF$ , por ello, la planta quedaría de la siguiente manera:

$$G(z) = \frac{1}{1 + 0,001034z}$$

Figura 5-2. Función de transferencia del circuito RC

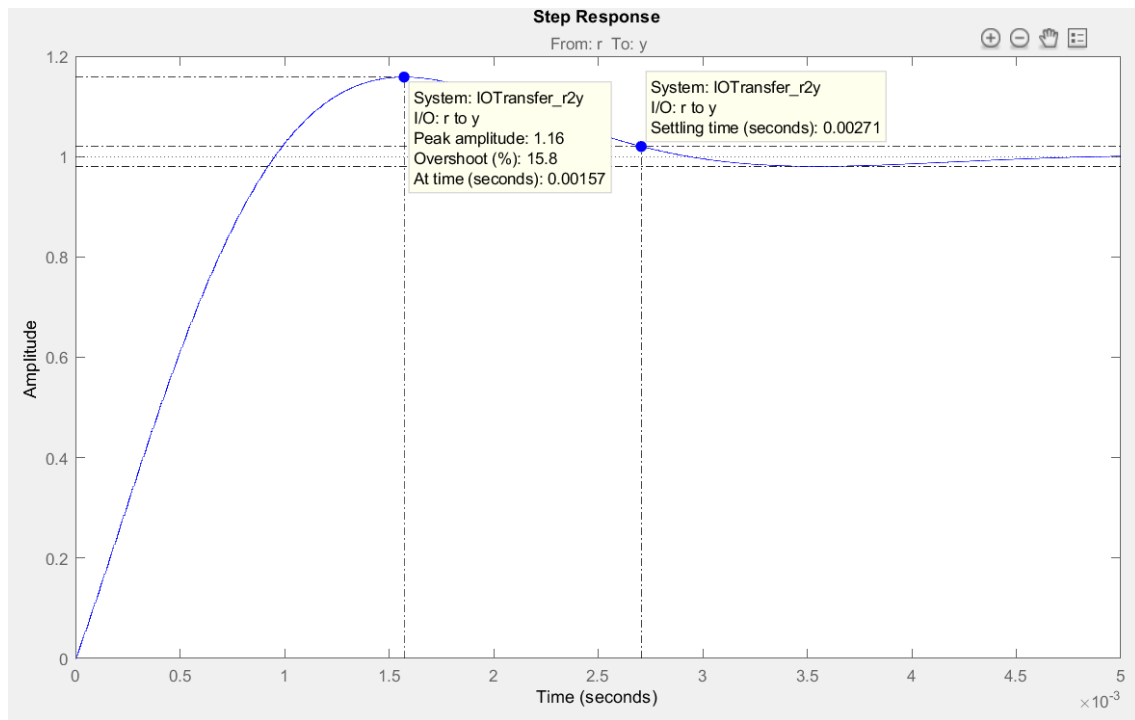
En la figura 5-2 se puede comprobar que en la planta hay un polo en  $z = \frac{-1}{RC}$ . El regulador se va a realizar con un periodo de muestro de 5120 ns, el diseño de este incluye un integrador, es decir un polo en 1, para que el error en régimen permanente sea nulo. Puesto que el orden del numerador debe ser menor o igual al denominador se puede añadir un cero cercano al polo de la planta para contrarrestar sus efectos. Por tanto, la función que el usuario debe crear tiene que ser de la forma:

$$G(z) = a * \frac{z - b}{z - c} = a * \frac{1 - bz^{-1}}{1 - cz^{-1}}$$

**Figura 5-3. Función general regulador RC**

La función creada en el caso de nuestro diseño es:

$R(z) = 1,2 * \frac{(1-0,984*z^{-1})}{(1-z^{-1})}$ , la respuesta al escalón de dicha función es la siguiente:



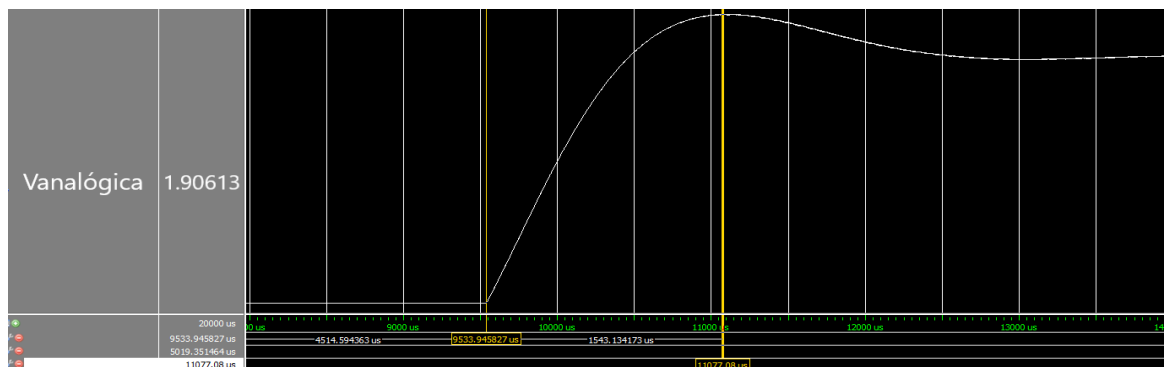
**Figura 5-4. Respuesta al escalón del circuito RC**

En la figura 5-4 se muestra el valor máximo de sobreoscilamiento que alcanza la señal, llegando a alcanzar un 16% más del valor de referencia, que en este caso es 1 debido a que todas las respuestas al escalón en Matlab son unitarias. También se puede comprobar que el tiempo que tarda hasta que la señal consigue estabilizarse es de 2,71 ms frente a los 4,05 ms que tarda la planta en estabilizarse.

Para comprobar que el sistema es correcto, se han realizado dos pruebas. Por una parte, se ha simulado el sistema junto a un modelo de la planta utilizando el simulador ModelSim. Por otra parte, se ha configurado el sistema final mediante la interfaz gráfica y se ha comprobado mediante un osciloscopio la tensión de salida del circuito RC regulado.

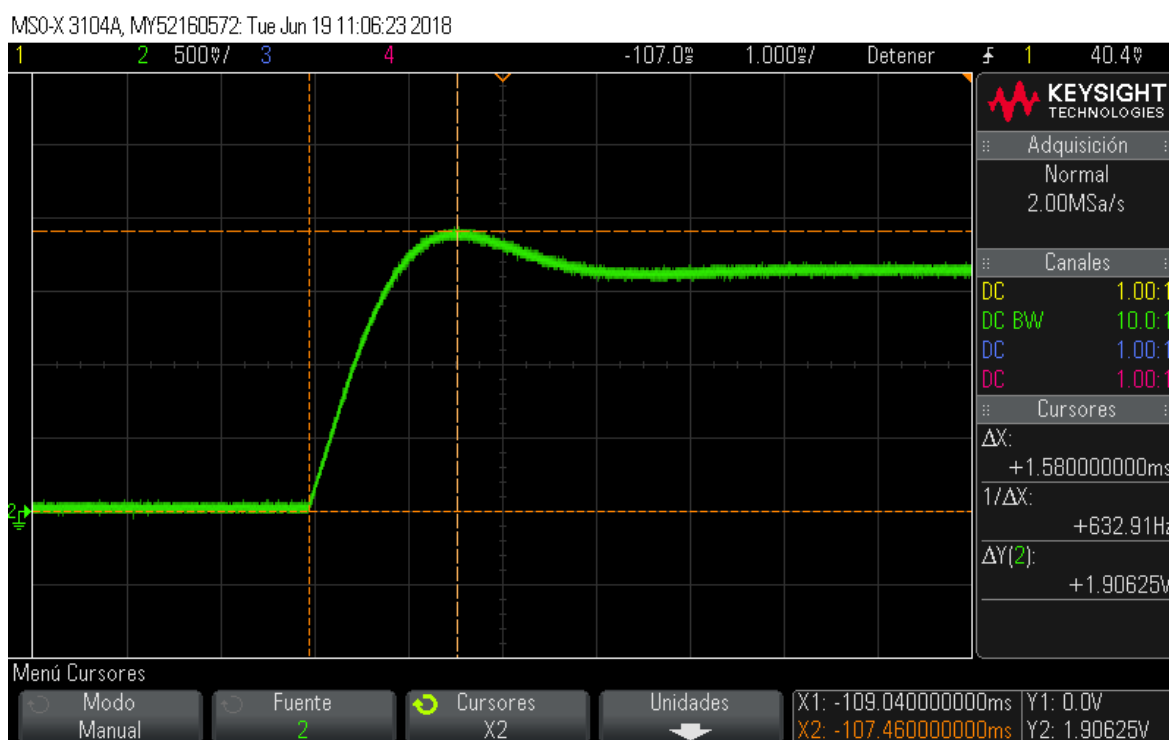
Para que todas las simulaciones sean lo más similares posibles el valor máximo de saturación se corresponderá con el valor máximo de voltaje, es decir, 3,3 V, además de asignar a los valores de las variables del regulador los mismos que en la fórmula del circuito RC. En ambos experimentos se ha probado un escalón de 0 a 1,65 V. Las capturas que no se ven con claridad han sido retocadas con el fin de mejorar su calidad de visualización, las originales se encuentran en el Anexo A.

En primer lugar, se realizará una comparación con el valor máximo de oscilamiento obtenido, mientras que en la segunda se procederá a comparar los tiempos de establecimiento.



**Figura 5-5. Simulación ModelSim circuito RC**

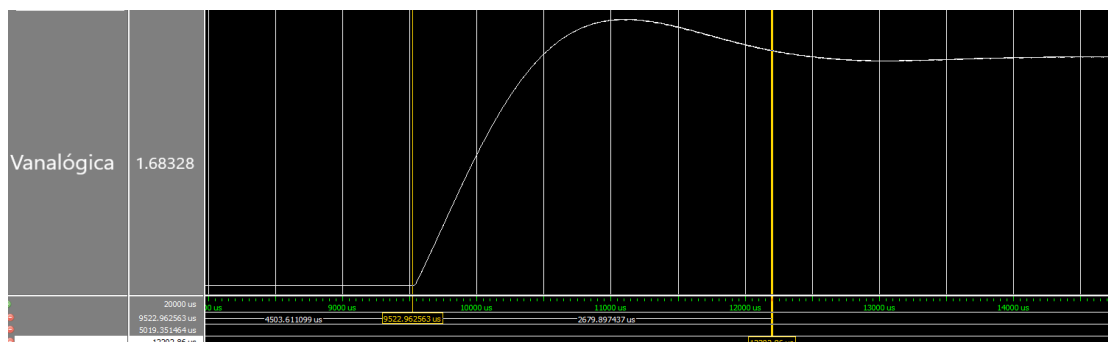
En la figura 5-5 puede observarse que el valor máximo al que llega la señal es de aproximadamente 1,90613 V, para ello ha tardado 1543,13 microsegundos. Cabe destacar que la referencia propuesta es de 1,65 V, correspondiéndose con la mitad del voltaje máximo posible. Por tanto, tendrá un sobreoscilamiento del 15,52% del valor final.



**Figura 5-6. Simulación real circuito RC**

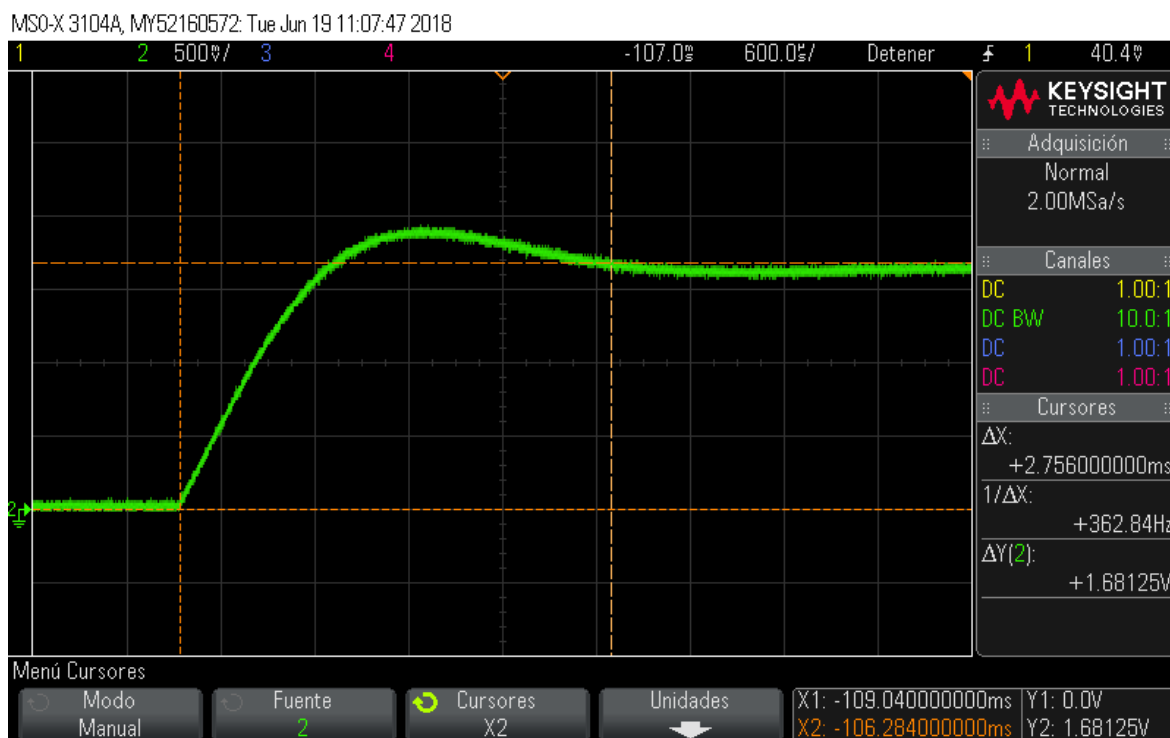
En la simulación real se puede comprobar cómo la forma de la onda, al igual que la simulación en ModelSim, es muy parecida a la mostrada mediante Matlab. El punto máximo de sobre oscilamiento es de 1,90125 V, es decir, un 15,53% del valor final, tardando 1,58 ms en llegar a él.

En *Sisotool* el tiempo de establecimiento es el 2% del valor final, por ello el valor utilizado es de 1,683 V como valor límite.



**Figura 5-7. Simulación ModelSim circuito RC**

Una vez se sabe el valor del voltaje para calcular el tiempo de establecimiento, se puede observar en la figura 5-7 que el tiempo de establecimiento es de 2,679 ms. En la simulación real se intenta buscar el mismo punto, se encuentra en el valor 1,68125 V, por ello el tiempo de establecimiento en la simulación real es de 2,756 ms.



**Figura 5-8. Simulación real circuito RC**

Una vez se han mostrado todas las simulaciones, se procede a realizar una comparación de los valores obtenidos mediante dichas gráficas.

	Tiempo de valor máximo de la señal	Porcentaje de sobre oscilación de la señal	Tiempo de establecimiento
Simulación Matlab	1,57 ms	15,80%	2,71 ms
Simulación ModelSim	1,54 ms	15,52%	2,68 ms
Simulación real	1,58 ms	15,53%	2,76 ms

**Tabla 5-1. Comparación circuito RC**

Una vez elaborada la tabla 5-1, se demuestra que el regulador creado es muy preciso, puesto que todos los valores que se han comparado son prácticamente idénticos. Para terminar con el circuito RC se realiza una comparación con los valores de salida y entradas del regulador en sus primeros ciclos. Para ello, se implementa el código para que el ADC siempre sea de valor 0, ejecutándose una comparación entre los valores con escalas y en decimal puro para ver cuántos decimales se pierden. Mediante este experimento se elimina la etapa de sensado, permitiendo que la aritmética del regulador pueda ser probada de forma fácil y fiable. La figura 5-9 muestra la simulación de los primeros ciclos de salida del regulador en ModelSim.

X_0	0	8192								
X_1	0		8192							
X_2	0			8192						
X_3	0				8192					
Y_0	0	4915	4993	5071	5149	5227	5305	5383	5461	5539
Y_1	0		4915	4993	5071	5149	5227	5305	5383	5461
Y_2	0			4915	4993	5071	5149	5227	5305	5383
Y_3	0				4915	4993	5071	5149	5227	5305
X_0_A_0	0	161054720								
X_1_A_1	0		-158482432							
X_2_A_2	0									
X_3_A_3	0									
Y_1_B_1	0		161054720	163610624	166166528	168722432	171278336	173834240	176390144	178946048
Y_2_B_2	0									
Y_3_B_3	0									
X_S	0	161054720	2572288							
Y_S	0		161054720	163610624	166166528	168722432	171278336	173834240	176390144	178946048
Data_in_X	0	161054720	2572288							
Data_out_X...	0	4915	78							
Data_in_Y	0		161054720	163610624	166166528	168722432	171278336	173834240	176390144	178946048
Data_out_Y...	0		4915	4993	5071	5149	5227	5305	5383	5461
Y_0	0	4915	4993	5071	5149	5227	5305	5383	5461	5539

Figura 5-9. Simulación ModelSim regulador circuito RC

PERIODO	0	1	2	3	4	5	6	7	8
ADC	0	0	0	0	0	0	0	0	0
Ref	8192	8192	8192	8192	8192	8192	8192	8192	8192
X0	8192	8192	8192	8192	8192	8192	8192	8192	8192
X1	0	8192	8192	8192	8192	8192	8192	8192	8192
X2	0	0	8192	8192	8192	8192	8192	8192	8192
X3	0	0	0	8192	8192	8192	8192	8192	8192
Y0	4915	4993	5071	5149	5227	5305	5383	5461	5539
Y1	0	4915	4993	5071	5149	5227	5305	5383	5461
Y2	0	0	4915	4993	5071	5149	5227	5305	5383
Y3	0	0	0	4915	4993	5071	5149	5227	5305
SEÑALES AUXILIARES									
X_0_A_0	161054720	161054720	161054720	161054720	161054720	161054720	161054720	161054720	161054720
X_1_A_1	0	-158482432	-158482432	-158482432	-158482432	-158482432	-158482432	-158482432	-158482432
X_2_A_2	0	0	0	0	0	0	0	0	0
X_3_A_3	0	0	0	0	0	0	0	0	0
Y_1_B_1	0	161054720	163610624	166166528	168722432	171278336	173834240	176390144	178946048
Y_2_B_2	0	0	0	0	0	0	0	0	0
Y_3_B_3	0	0	0	0	0	0	0	0	0
SUMAS Y RESULTADO FINAL									
PERIODO	0	1	2	3	4	5	6	7	8
XS	4915	78	78	78	78	78	78	78	78
YS	0	4915	4993	5071	5149	5227	5305	5383	5461
Y0	4915	4993	5071	5149	5227	5305	5383	5461	5539

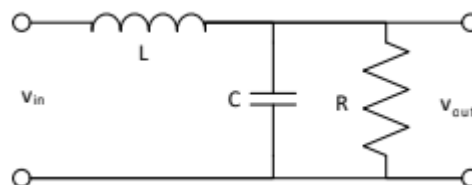
Figura 5-10. Simulación sin decimales Excel regulador circuito RC

PERIODO	0	1	2	3	4	5	6	7	8
ADC	0	0	0	0	0	0	0	0	0
Ref	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X1	0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X2	0	0	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X3	0	0	0	0,5	0,5	0,5	0,5	0,5	0,5
Y0	0,6	0,6096	0,6192	0,6288	0,6384	0,648	0,6576	0,6672	0,6768
Y1	0	0	0,6096	0,6192	0,6288	0,6384	0,648	0,6576	0,6672
Y2	0	0	0	0,6096	0,6192	0,6288	0,6384	0,648	0,6576
Y3	0	0	0	0	0,6096	0,6192	0,6288	0,6384	0,648
SEÑALES AUXILIARES									
X_0_A_0	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6
X_1_A_1	0	-0,5904	-0,5904	-0,5904	-0,5904	-0,5904	-0,5904	-0,5904	-0,5904
X_2_A_2	0	0	0	0	0	0	0	0	0
X_3_A_3	0	0	0	0	0	0	0	0	0
Y_1_B_1	0	0,6	0,6096	0,6192	0,6288	0,6384	0,648	0,6576	0,6672
Y_2_B_2	0	0	0	0	0	0	0	0	0
Y_3_B_3	0	0	0	0	0	0	0	0	0
SUMAS Y RESULTADO FINAL									
PERIODO	0	1	2	3	4	5	6	7	8
XS	0,6	0,0096	0,0096	0,0096	0,0096	0,0096	0,0096	0,0096	0,0096
YS	0	0	0,6096	0,6192	0,6288	0,6384	0,648	0,6576	0,6672
Y0	0,6	0,6096	0,6192	0,6288	0,6384	0,648	0,6576	0,6672	0,6768
Y0*2*escalaY	4915,2	4993,8432	5072,4864	5151,1296	5229,7728	5308,416	5387,0592	5465,7024	5544,3456

**Figura 5-11. Simulación con decimales Excel regulador circuito RC**

En estas tablas se muestran los valores ciclo por ciclo de las entradas  $X_{0-3}$  y las salidas  $Y_{1-3}$ . Al igual que en la figura 3-10, la suma de todas las salidas y entradas es la salida total del regulador,  $Y_0$ . Por ello el valor de  $Y_0$  se utiliza para recalcular su valor un ciclo después como se puede observar en la tabla. El primer ciclo solo tendrá valores de entrada debido a que la primera salida es 0, una vez se obtiene la primera salida los valores irán cambiando hasta conseguir la referencia pedida. Visualizando las tablas, los valores son prácticamente iguales. De hecho, los valores sin decimales son idénticos en todos los ciclos a los mostrados en la simulación de ModelSim; en cambio, a la hora de trabajar con decimales se adquiere más precisión y por ello los valores no son idénticos, pero sí muy parecidos. Por tanto, se puede concluir que para el circuito RC, el regulador ejecutado en el sistema universal diseñado en este TFG es realmente preciso debido a los excelentes resultados obtenidos.

## 5.2 Circuito RLC



**Figura 5-12. Circuito RLC**

$$G(w) = \frac{V_{out}}{V_{in}} = \frac{Z_c // R}{Z_l + Z_c // R} = \frac{\frac{R}{jwRC + 1}}{jwL + \frac{R}{jwRC + 1}} = \frac{R}{j^2 w^2 LRC + jwL + R} = \frac{1}{-w^2 LC + w \frac{L}{R} + 1} \rightarrow G(z) = \frac{1}{-LCz^2 + \frac{L}{R}z + 1}$$

Donde los valores son:  $R=4,7K\Omega$ ,  $L=5.6mH$  y  $C=220nF$ . Por tanto, la planta quedaría de la siguiente manera:

$$G(z) = \frac{1}{-1,232 * 10^{-9} z^2 + 1.19 * 10^{-6} z + 1}$$

**Figura 5-13. Función de transferencia circuito RLC**



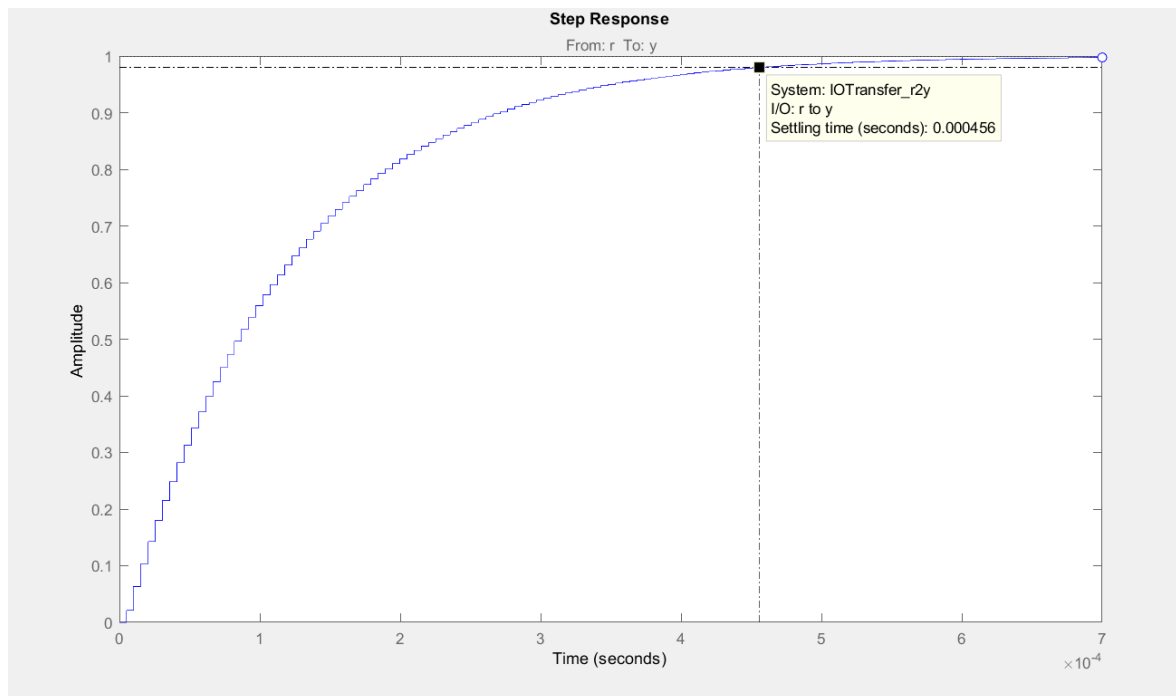
Como se puede observar en la figura 5-13, la planta contiene dos polos. En el diseño que se va a crear primero hay que introducir un polo en 1, es decir, un integrador, para que el error en régimen permanente sea 0. Como la planta contiene dos polos se añaden dos ceros con el fin de contrarrestar dichos polos. Por último, se añade un polo debido a que el orden del denominador debe ser igual o mayor al del numerador, normalmente ese polo se coloca en 0 para no interferir con el otro polo, por ello el sistema RLC tendrá la siguiente forma:

$$R(z) = a \frac{(z - b)(z - c)}{(z - d)(z - e)}$$

**Figura 5-14. Función general regulador RLC**

Por tanto, la fórmula de un regulador para un circuito RLC como se ha expuesto anteriormente tendría dos polos y dos ceros, con el fin de contrarrestarse. El regulador que se propone en este trabajo será el siguiente:

$R(z) = 2 \frac{(z-0.9869+0.145i)(z-0.9869-0.145i)}{z(z-1)}$ , dicha función tiene dos ceros conjugados en 0.9869 con parte imaginaria en 0,145 y dos polos, uno en 0 y el otro en 1. La respuesta al escalón calculada mediante Matlab a dicho regulador es la mostrada la figura 5-15:



**Figura 5-15. Respuesta al escalón circuito RLC**

En la figura anterior se muestra cómo el circuito no presenta sobreoscilamiento, a diferencia del circuito anterior; por ello su comparación con las demás simulaciones será más escueta. Al igual que en todas las respuestas al escalón de Matlab, el valor final de la onda es de 1. El tiempo de establecimiento es de 0,456 ms, mientras que el de la planta era de 8,06 ms, con lo cual se ha reducido mucho el tiempo de respuesta. Al igual que en las simulaciones anteriores, el valor máximo de saturación es igual al máximo voltaje, 3,3V. Para empezar con la comparación se muestran tanto la simulación en ModelSim como una

real. Al igual que en el circuito RC, el punto para calcular el tiempo de establecimiento se ha resuelto de la misma manera. Por tanto, el valor buscado es 1,617 V, por ello el tiempo de establecimiento es de 0,472 ms.



Figura 5-16. Simulación real circuito RLC

En la figura 5-17 se muestra la respuesta al escalón del regulador mediante una simulación en ModelSim, se observa como la forma de la onda es parecida a la de la simulación real, pero con más oscilación al final, el tiempo de establecimiento en esta simulación es de 0,774 ms.

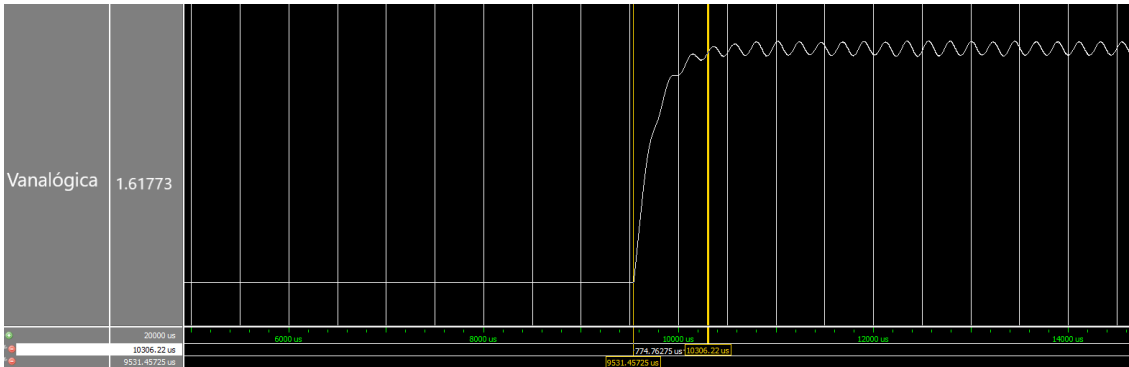


Figura 5-17. Simulación ModelSim circuito RLC

Tipo de simulación	Tiempo de establecimiento
Simulación Matlab	0,456 ms
Simulación ModelSim	0,774 ms
Simulación real	0,472 ms

Tabla 5-2. Comparación circuito RLC

En la tabla comparativa podemos ver como en la simulación mediante ModelSim el tiempo de establecimiento ha sido mayor, mientras que la simulación real tiene un tiempo muy parecido a la simulación teórica.

Una vez se ha realizado la comparación de las ondas de salida, es el turno de ejecutar la comparación de los primeros valores de salida del regulador durante los primeros ciclos de simulación. Al igual que en caso del circuito RC, la comparación se va a realizar entre Excel y ModelSim.

PERIODO	0	1	2	3	4	5	6	7	8	9	10
ADC	0	0	0	0	0	0	0	0	0	0	0
Ref	8192	8192	8192	8192	8192	8192	8192	8192	8192	8192	8192
X0	8192	8192	8192	8192	8192	8192	8192	8192	8192	8192	8192
X1	0	8192	8192	8192	8192	8192	8192	8192	8192	8192	8192
X2	0	0	8192	8192	8192	8192	8192	8192	8192	8192	8192
X3	0	0	0	8192	8192	8192	8192	8192	8192	8192	8192
Y0	8192	246	451	656	861	1066	1271	1476	1681	1886	2091
Y1	0	8192	246	451	656	861	1066	1271	1476	1681	1886
Y2	0	0	8192	246	451	656	861	1066	1271	1476	1681
Y3	0	0	0	8192	246	451	656	861	1066	1271	1476
SEÑALES AUXILIARES											
X_0_A_0	134217728	134217728	134217728	134217728	134217728	134217728	134217728	134217728	134217728	134217728	134217728
X_1_A_1	0	-264404992	-264404992	-264404992	-264404992	-264404992	-264404992	-264404992	-264404992	-264404992	-264404992
X_2_A_2	0	0	133545984	133545984	133545984	133545984	133545984	133545984	133545984	133545984	133545984
X_3_A_3	0	0	0	0	0	0	0	0	0	0	0
Y_1_B_1	0	268435456	8060928	14778368	21495808	28213248	34930688	41648128	48365568	55083008	61800448
Y_2_B_2	0	0	0	0	0	0	0	0	0	0	0
Y_3_B_3	0	0	0	0	0	0	0	0	0	0	0
SUMAS Y RESULTADO FINAL											
PERIODO	0	1	2	3	4	5	6	7	8	9	10
XS	8192	-7946	205	205	205	205	205	205	205	205	205
YS	0	8192	246	451	656	861	1066	1271	1476	1681	1886
Y0	8192	246	451	656	861	1066	1271	1476	1681	1886	2091

Figura 5-18. Simulación sin decimales Excel regulador circuito RLC

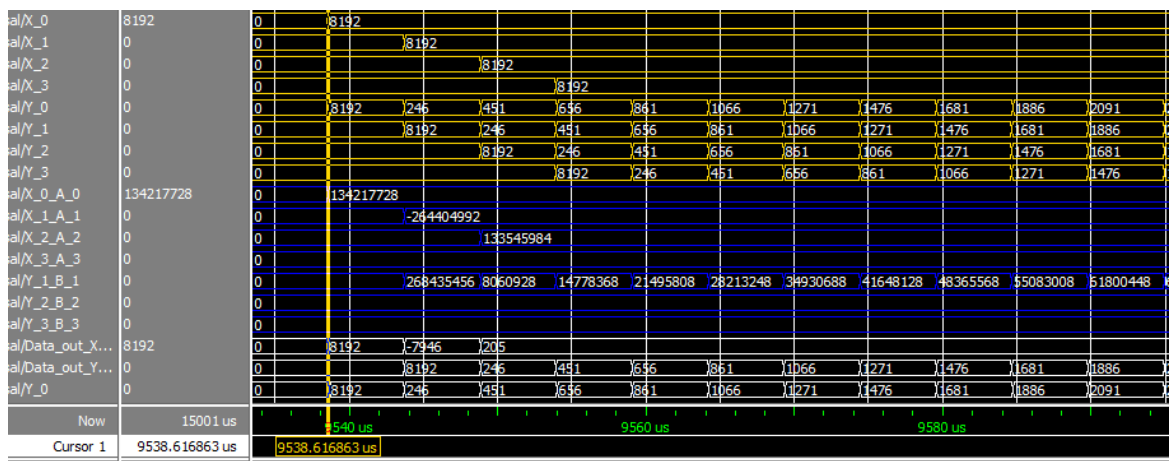


Figura 5-19. Simulación ModelSim regulador circuito RLC

PERIODO	0	1	2	3	4	5	6	7	8	9	10
ADC	0	0	0	0	0	0	0	0	0	0	0
Ref	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X1	0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X2	0	0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
X3	0	0	0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
Y0	1	0,03	0,055	0,08	0,105	0,13	0,155	0,18	0,205	0,23	0,255
Y1	0	1	0,03	0,055	0,08	0,105	0,13	0,155	0,18	0,205	0,23
Y2	0	0	1	0,03	0,055	0,08	0,105	0,13	0,155	0,18	0,205
Y3	0	0	0	1	0,03	0,055	0,08	0,105	0,13	0,155	0,18
SEÑALES AUXILIARES											
X_0_A_0	1	1	1	1	1	1	1	1	1	1	1
X_1_A_1	0	-1,97	-1,97	-1,97	-1,97	-1,97	-1,97	-1,97	-1,97	-1,97	-1,97
X_2_A_2	0	0	0,995	0,995	0,995	0,995	0,995	0,995	0,995	0,995	0,995
X_3_A_3	0	0	0	0	0	0	0	0	0	0	0
Y_1_B_1	0	1	0,03	0,055	0,08	0,105	0,13	0,155	0,18	0,205	0,23
Y_2_B_2	0	0	0	0	0	0	0	0	0	0	0
Y_3_B_3	0	0	0	0	0	0	0	0	0	0	0
SUMAS Y RESULTADO FINAL											
PERIODO	0	1	2	3	4	5	6	7	8	9	10
XS	1	-0,97	0,025	0,025	0,025	0,025	0,025	0,025	0,025	0,025	0,025
YS	0	1	0,03	0,055	0,08	0,105	0,13	0,155	0,18	0,205	0,23
Y0	1	0,03	0,055	0,08	0,105	0,13	0,155	0,18	0,205	0,23	0,255
Y0*2^escalaY	8192	245,76	450,56	655,36	860,16	1064,96	1269,76	1474,56	1679,36	1884,16	2088,96

**Figura 5-20. Simulación con decimales Excel regulador circuito RLC**

Al igual que en el circuito RC, la simulación con decimales difiere levemente en los términos intermedios, pero en un factor de decimales bajo. Esto es debido al uso de 17 bits en los coeficientes y variables del diseño, provocando la leve pérdida de resolución. Sin embargo, la precisión del sistema sigue siendo muy alta, mientras que la velocidad y área del sistema están optimizadas.

Una vez se han realizado las simulaciones correspondientes, observamos los recursos utilizados por la simulación, al utilizar coma fija parametrizable el uso será menor que utilizando coma flotante.

	Usados	Disponibles	Utilizados (%)
FF	574	3840	14%
LUTs	3108	3840	80%
Mult18x18	9	12	75%

**Tabla 5-3. Recursos utilizados**

Mediante la tabla 5-3 se puede comprobar que no se llega a utilizar el 100% de los recursos que el sistema proporciona, de hecho el porcentaje de Flip Flops es bastante pequeño. En cambio, al tener que realizar numerosas multiplicaciones en el regulador, el número de LUTs y multiplicadores están entorno al 80%.

## 6 Conclusiones y trabajo futuro

---

### 6.1 Conclusiones

Los reguladores digitales cada vez son más usados en los sistemas de control gracias a sus numerosas ventajas a la vez que su precio va reduciéndose. Para lograr ejecutar un control digital se utiliza, entre otros dispositivos, FPGAs debido a su velocidad y capacidad de creación de lazos de control complejos con altas velocidades de actualización.

Con la intención de optimizar los recursos y velocidad del sistema, es habitual utilizar coma fija como el tipo de aritmética del regulador. Este tipo de aritmética es compleja y dicha dificultad se une a la propia de codificar en VHDL, lenguaje más complejo que los lenguajes software. Por ello, el aprendizaje de los conceptos de teoría de control unidos a la codificación de sistemas complejos en VHDL, hace que el aprendizaje sea costoso.

Este TFG se ha centrado en el diseño e implementación de un regulador universal, cuya configuración (constantes, rango numérico de las señales, etc.) pueda ser configurado en tiempo real, sin tener que resintetizar el código que contiene. Para ello se ha creado el sistema mediante un nuevo tipo de dato numérico llamado coma fija parametrizable. Este término combina la alta velocidad y bajos recursos de la coma fija y la adaptabilidad de la coma flotante. Para que este sistema se adapte correctamente a cualquier valor se ha creado un sistema de desplazamientos numéricos que permite interpretar correctamente cada número según la posición de la coma del mismo.

Una vez se consigue una implementación capaz de adaptarse a cualquier valor se ha diseñado una aplicación con la que configurar esos valores. En este proyecto se crea una interfaz gráfica con la que el usuario puede introducir los valores y cambiarlos cuando él desee. Para ello se ha dotado al sistema de una comunicación serie para poder recibir la información desde un ordenador.

Una vez se ha implementado todo el código necesario para su funcionamiento se introducen los circuitos los cuales queremos regular, en este trabajo, de tipo RC y RLC. Al ser un trabajo dedicado a docencia los alumnos primero tendrán que realizar una primera simulación teórica del regulador creado mediante la herramienta de Matlab *Sisotool*. Una vez se comprueba que los valores son correctos, se introducen en la interfaz gráfica y se observa la salida correspondiente mediante un osciloscopio.

Para comprobar el correcto diseño realizar en este TFG, se han realizado numerosos experimentos teóricos y prácticos, demostrando que las dinámicas obtenidas, la respuesta del regulador, y la respuesta del sistema en lazo cerrado, son las correctas. Por tanto, se puede concluir que este Trabajo Fin de Grado permitirá la implantación de un nuevo sistema de aprendizaje en la asignatura Sistemas de control, permitiendo que los estudiantes se centren en la teoría de control y no la implementación HDL del regulador diseñado.

## **6.2 Trabajo futuro**

Como líneas de futuro se propone la elaboración de un código que obtenga mayor resolución en las constantes del regulador. Esto se obtiene cambiando la implementación del regulador, para conseguir dicho resultado las señales de 17 bits tendrán que ser más grandes. El aumento de las variables tendrá efectos negativos en el área y velocidad del diseño, pero éstos pueden ser estudiados para llegar a un punto de equilibrio.

También se podría mejorar el diseño de la interfaz gráfica añadiendo explicaciones sobre los sistemas de control para que así el estudiante pueda aprender más gracias a la aplicación. Además de realizar más pruebas con otros circuitos aparte de los realizados en las prácticas.

## Referencias

- [1] K. Kołek et al., "Laboratory real-time systems to facilitate automatic control education and research," Proceedings of the International Multiconference on Computer Science and Information Technology, Wisla, 2010, pp. 805-812.
- [2] H. Wenshan, Z. Hong and L. Guoping, "Web-based 3D laboratory for control engineering education," Proceedings of the 31st Chinese Control Conference, Hefei, 2012, pp. 5820-5825.
- [3] Zilouchian, "The development of a real-time digital computer control laboratory for electrical engineering education," Southeastcon, Proceedings., IEEE, Birmingham, AL, pp. 457-460 vol.1.
- [4] F. Lopez-Colino and A. Sanchez and M. S. Martinez-Garcia and M. Murillo and A. de Castro and G. Glez-de-Rivera and J. Garrido, "Development of an online platform for hardware-based laboratories in Engineering studies," 2016 Conference on Design of Circuits and Integrated Systems (DCIS), Granada, 2016, pp. 1-5.
- [5] D. Filahi, N. Bennis and S. E. Hani, "Generic client-server interfaces for real-time remote automatic control laboratory," 2017 International Conference on Electrical and Information Technologies (ICEIT), Rabat, 2017, pp. 1-7.
- [6] Uran Suzana Jezernik Karela "Virtual laboratory for relative control design experiments" IEEE Transactions on Education vol. 51 no. 1 pp. 69-75 February 2008.
- [7] National Instruments. <http://www.ni.com/es-es.html>.
- [8] A. Abu-Aisheh F. Farahmand "LabVIEW-based Integrated Virtual Learning Platform" 10th IEEE International Conference on Advanced Learning Technologies 2010.
- [9] Losada, C, Espinosa, F, Rodríguez, JM, Rodríguez, FJ, Marron, M, "Entorno de diseño electrónico para implementación de control remoto con tarjeta de evaluación miniDK2 para formación en ingeniería de control electrónico de alumnos de grado."
- [10] SpCard. SP Control Technologies, Junio 2018.  
<https://www.thespcard.com/uploads/9/4/0/7/94079977/sptool.pdf>
- [11] NI myDAQ. National Instruments, Junio 2018. <http://www.ni.com/es-es/shop/select/mydaq-student-data-acquisition-device>
- [12] NI myDAQ. National Instruments, Junio 2018. <https://www.ni.com/es-es/shop/select/myrio-student-embedded-device>
- [13] María Murillo, "Control digital de convertidores conmutados mediante LabVIEW". Trabajo Fin de Máster, Universidad Autónoma de Madrid, 2016.
- [14] Matlab, "www.mathworks.com," 2013.
- [15] Ogata, Katsuhiko. "Ingeniería de Control Moderna", Pearson, Prentice Hall, 2003, 4ª Ed., Madrid.
- [16] Ramon Neco, Óscar Reinoso, Nicolás García, Rafael Aracil, "Apuntes de Sistemas de Control", Editorial Club Universitario, 2003.
- [17] Xilinx Application Note XAPP467. Disponible online. Junio 2018.  
[https://www.xilinx.com/support/documentation/application\\_notes/xapp467.pdf](https://www.xilinx.com/support/documentation/application_notes/xapp467.pdf)
- [18] A. Sanchez, A. de Castro & J. Garrido, "A comparison of simulation and hardware-in-the-loop alternatives for digital control of power converters", in IEEE Transactions on Industrial Informatics, vol. 8, no. 3, pp. 491-500, ago 2012.





## Glosario

---

PID	Proporcional-Integral-Derivativo
FPGA	<i>Field-Programmable Gate Array</i>
VHDL	<i>VHSIC Hardware Description Language</i>
PWM	<i>Pulse Width Modulation</i>
RC	<i>Resistencia-Condensador</i>
RLC	<i>Resistencia-Bobina-Condensador</i>
DSP	<i>Digital Signal Processor</i>
ADC	<i>Analog-to-Digital Conversion</i>
FF	<i>Flip Flop</i>



# Anexos

## A Capturas originales

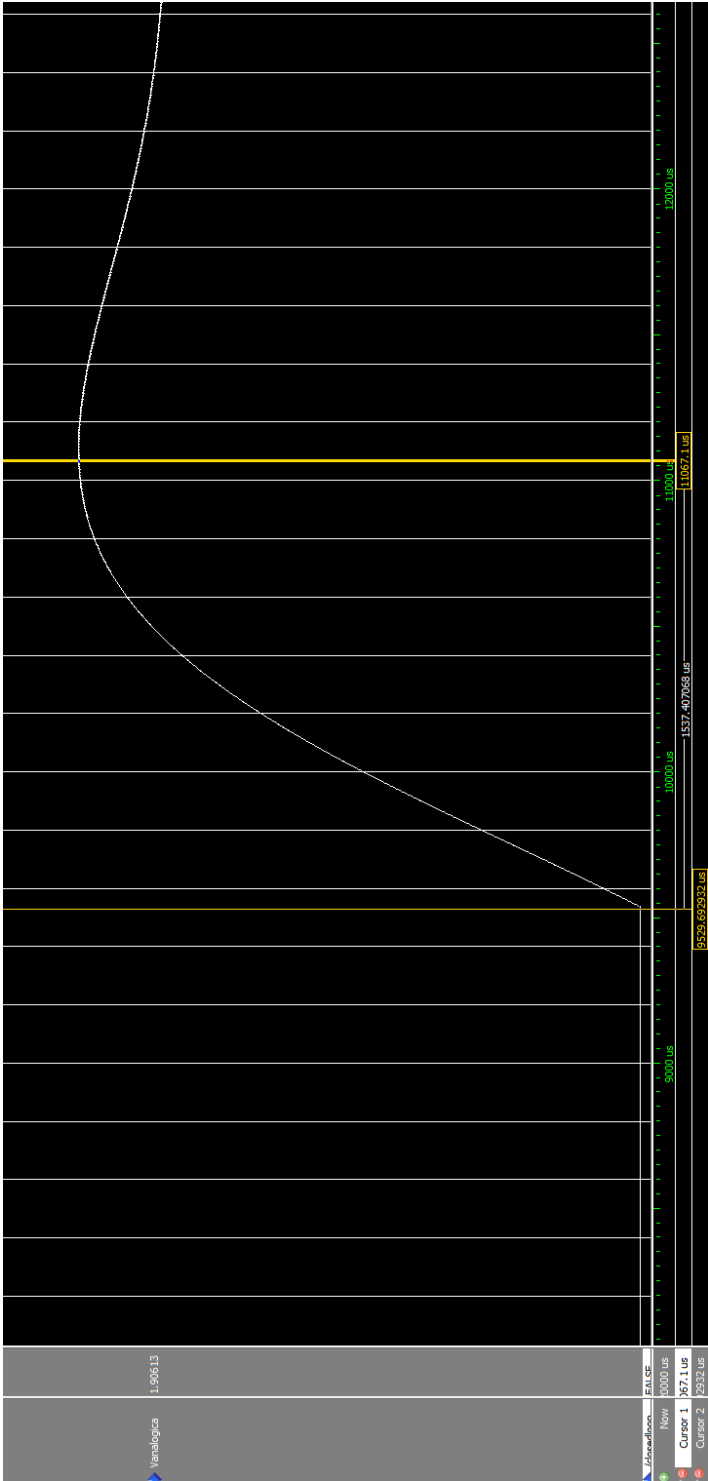


Figura A-1. Simulación ModelSim sobreoscilamiento circuito RC



Figura A-2 Simulación ModelSim circuito RC



Figura A-3 Simulación ModelSim RLC